avgust 1988 časopis za popularizaciju informatike i računarstva

računari 41

cena 2000 dinara



imetak

turbo pascal 4.0

komercijalni softver

modbase

wordperfect 5.0

emulatori

pc na atariju

periferijska oprema

hp laserjet series II

softverski podsetnik

game maker/spektrum
music system/amstrad

flight simulator/atari st

tehnike programiranja

algoritmi

razbarušeni sprajtovi

9 778603 527279

računari su uvek aktuelni!

Ono što nema u ovom broju, potražite u starim brojevima:

- *novi uređaji, uporedni prikazi, testovi*
- *tehnika programiranja, algoritmi, biblioteka programa*
- *komercijalni softver, softverski podsetnici*
- *vesti, domaća i strano tržište*
- *klub programara, HELP, bajtovi lična priroda, pet plus*
- *akcija: samogradnja računara „tim 011“ samogradnja robota*

















računari 34

Operativni sistemi *UNIX*
Algoritmi *Digitalna računalnika*
Komercijalni softver *Borlandova visoka C*
Serije Metematički softver — metodi aproksimacija Put u središte ROM-a — Simulatori

računari 35

Hardver *tim 011 Multisinhroni monitori matematički koprocesori*
Komercijalni softver *QUICK Basic*
Operativni sistemi *UNIX*
Serije Put u središte ROM-a — disasembler
Mali kućni roboti Crtanje na bejziku Stono izdevaštvo

računari 36

Hardver *Psion Organizer II 100 PC klonova*
Tehnike programiranja *GUS liste CoreWar*
Serije UNIX Mali kućni roboti Stono izdevaštvo
Crtanje na bejziku

računari 37

Hardver Pekardov i bombončići Bubble-jet štampači
Komercijalni softver Turbo Pascal 4.0 Meke
Algoritmi liste
Serije Stono izdevaštvo Napravi kućnog robota.
Put u središte ROM-a — monitor
Umetak *dBASE III plus*

računari 38

Hardver *EGA kertice* komodor *MPS 1200 ko-procesor 80387*
Komercijalni softver *Smalltalk/V*
Serije Stono izdevaštvo UNIX
Umetak. *Lotus 1 2 3*

računari 39

Hardver *AT beba*
Komercijalni softver *MASM Majkrosoftovi novi-teti module-2*
Algoritmi *liste inteligentno pretraživanje*
Serije *Napravi melog robota*
Umetak MS DOS 3 30

računari 40

Komercijalni softver WordPerfect 5 0 PC TOOLS DELUXE MuMATH CED
Operativni sistemi VMS
Algoritmi liste inteligentno pretraživanje
Softverski podsetnik Art Studio GEOS 1 3
Umetak Xerox Venture Publisher

NARUDŽBENICA

Molim vas da mi pošaljete sledeća brojeve Računara

34 35 36 37 38 39 40

Odgovarajući iznos uvećan za troškove pakovanje i poštarine platiću poštaru prilikom preuzimanje pošiljke

Ime i prezime .....

Mesto .....

Adresa .....

U svakom broju opširan priručnik

Mala PC biblioteka
broj priručnik
37 dBASE III plus
38 Lotus 1 2 3
39 MS DOS 3 30
40 Xerox Venture Publisher

računari 41

časopis za popularizaciju informatike i računarstva izlazi jednaput mesečno
YUISSN 0352-7271
izdaje RIGZ
avgust 1988
cena 2000 dinara

Izdaje
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar vojvode Mišića 17

Generalni direktor
Dobrosav Petrović

Zamenik generalnog direktora
Antun Martić

Glavni i odgovorni urednik časopisa „Galaksija"
Stanko Stojiljković

Glavni i odgovorni urednik
Jovan Ragasek

Stručne redakcije
Žarko Barberski Voja Gašić Slobodan Perović Dejan Ristanović
Jovan Skuljan prof dr Dušan Sivić
Nevanka Spalević Zoran Životić Anđelko Zgoralec

Sekretar redakcije
Zorka Simović

Tehnički urednik
Dušan Mijatović

Pomoćnik tehničkog urednika
Mirko Popov

Marketing
Sergija Marčenko

Stalni saradnici
Neda Aleksić Slobodanka Ast Žarko Berberski Viktor Cerovski Zoran Cvijetić Ninoslav Čabrić Voja Gašić Željko Jurić Blažimir Mišo Zoran Obradović, Slobodan Perović Miodrag Potkonjak Aleksandar Radovanović Dejan Ristanović Jelena Rupnik Duško Sević Dušan Sivić Jovan Skuljan Nevanka Spalević Vlada Stojiljković Saša Svitlica Milan Tadić, Žarko Vukosavljević, Anđelko Zgoralec Zoran Životić

Izdavački savet „Galaksije"
Dr Rudi Dobijadi prof dr Branislav Dimitrijević (predsednik) Radovan Drašković Tanasije Gavranović Živorad Gligić Esad Jakupović Velizar Masleć Nikola Pajić Željko Perunović prof dr Momčilo Ristić Vlada Ristić dr inž Milorad Teofilović Vidojko Veličković Velimir Vasović Milivoje Vuković

Adresa redakcije
11000 Beograd Bulevar vojvode Mišića 17/III

Telefoni
650-181 (sekretarijat)
653-748 (redakcija)
650-528 (prodaja)
761-793 (propaganda)

Rukopisi se ne vraćaju

Štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar vojvode Mišića 17

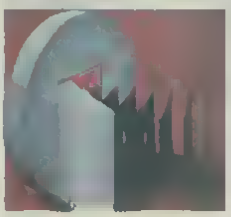
Pretplata u zemlji
Za jednu godinu 24 000 –
Za šest meseci 12 000 –

Pretplata za inostranstvo
Za jednu godinu 48 000 –
Za šest meseci 24 000 –

Inostranstvo u devizama
Za jednu godinu 18 USD 34 DM 28 CHF 10 GBP 114 FRF
Za šest meseci polovina od navedenog iznosa







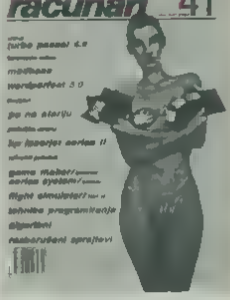
Sadržaj

3) Šta ima novo
10) Proširenja
Četiri asa
12) Naš test/HP LaserJet S II
Laserski džet-set
14) Računari u razgovoru /Dragan Kopunović
Znanja za uvek ispiati
16) Komarcijalni softver-/WordPerfect 5.0
Niko nije WordPerfectan
21) Komarcijalni softver /ModBase
Moduli za moduiu
23) Emulatori/„atari ST"
Računar sa hiljadu lica
26) Softverski podsetnik
28) Operativni sistemi/VMS
Na, nemoj mi prići
31) Umatak
Turbo Pascal 4.0
47) Dejanove pitalice
48) Algoritmi
Koren, stabio, list
50) Tahnika programiranja /aritmatička operacija
Osnovna računaka radnje
55) Tahnika programiranja /inteligentno pretraživanja
Uz malu pomoć pamati
53) Tahnika programiranja /Quick Radix Sort
Brže od najbržeg
54) Tahnika programiranja
Brzi Gonzalas
58) Klub programara
82) Help
65) Razbarušeni sprajtovi



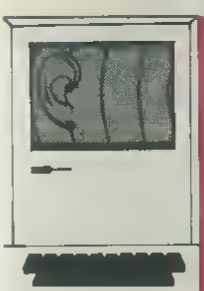






Iako je letnje vreme Računari se i dalje čitaju a za hlađenje se koriste diskete. Uz dunca nga greje i menekenka Olja Jelić a blendu zatvara Zoran Tadić Dirigentaiku palicu držao je Stanislav Berđević

Izvinjavamo se Brasi i Zoranu što nisu pomenuti u prošlom broju (ugievnom zbog Ksenijinog šarma).

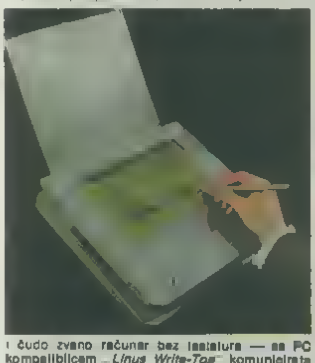


Šta ima novo

Nove tehnologije

Računar baz tastature

Perfabi računarima mnoge smatra tastature — ili je suviše male da bi se na njoj ozbiljno radilo ili je suviše velika da biamo je legitili unesokoita. Zahvaljujući firmi *Linus Technologies* (američki telefon (703)478-1500) od nedavno je na tržištu



i čudo zvano računar bez tastature — za PC kompatibilicem *Linus Write-Top* komunicirate posredstvom specijalne olevka kojom pišete po ekranu tvrdi se da će računar prepoznati i najneitljiviji rukopis. Ovo čudo tehnika zasnovano na oblčnom 8088 i dopunjeno disk jedinicom teži oko tri kilograma i tri kilodolara. Neverna Toma jasno mogu da dokupe i opelcnu tastaturu!

Masovna memorija

Tri računara na jadnom disku

Težnja softverskih kuća da svoje igre implementiraju na raznim računarima navela je britanku firmu *Mandarin Software* da razvija specijalan disk format koji obezbeduje da se na jednu disketu od 3 inče snimi program za spekirum plus 3 amstrad 6128 i amstrad PCW: svaki računar će videti i samim tim učitavati samo onu varziju programa koje mu odgovara. Obezbedano je ćak i korišćenje druga strana diskete za smeštanje ekrana koji su takoda upotrebljivi na sva tri računara.

Hvala firmi *Mandarin Software* ali bi bilo mnoge lapše kada bi plus 3 6128 i PCW koristili isti format upise na disk — sva tri računara na kraju krajeva prodaja iata firma!

Komarcijalni softver

Nortonovi vodiči

Softverska kompanija Pitera Nortona nija slavna samo po paketu Norton Utilities uz brojna knjiga nedavno je pramovisen i softverski podsetnik u više tomova zvani *Instant Access*. Radi se zaprevo o razidaninom programu (rezidentni program je ono šlo jada memorlju u ovom slučaju pojedena mamorija ne prefazi 68 kilokejta) koji će vam u svakoj kritičkoj situaciji pružiti sva potrebna obavaštenja o MS DOS-u OS/2 bajziku C-u asemblaru *Turbo Pascalu*. Svaki od podsetnika košta po 55 funti dok se pri kupovini dva paketa dobija popust — svega 100 funti.

Komarcijalni softver

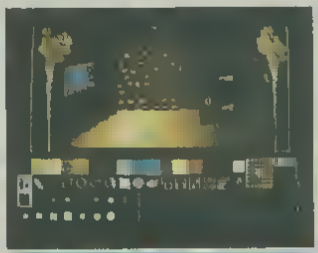
Hisoftovi etari noviteti

Sećate li se britanske kuće HiSoft koja se proslavila ozbiljnim programima za spekirum? Ozbiljni programi se i dalja pišu ali su mašine promanjana — težište je na Atarijevoj seriji ST računara. Mogućnost očuvanja sadržaja RAM diska posle *reset*-a bi svakako bila sasvim dovoljna da opravda cenu paketa *Saved! 2.0*. Spasonosni paket medulim nudi i mnoge druga stvari automatsko kopiranje datoteke ili čitavih direktorijuma na RAM disk po uključenju računara, korišćenja RAM diska za programe i floplja za podatke korišćenja svih komandi GEM-a iz aplikativnih programa i to posredstvom tastatura ili miša povraćaj grešikom obrisanih datoteka. HiSoftov britanski telefon je (&25)718181.

Igre

Prava grafika

Sirotl PC-jevci opremljeni EGA-ame i Herkuleslma smaliraju de ja grafike u igri *Leisure Suit Larry in the Land of the Lounge Lizards* nešto naročita. Možda zato šlo još nisu videli naku arkadnu avanturu za atari ST ili amigu — šta kažete na ova prizora iz Rainbirdove igre *Legend of the Sword* (atari ST) i infocomeve igre *Chamber of the Guardian* (amiga)?

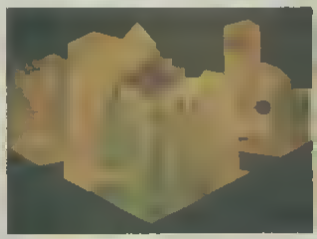




Inostrani život

Valika jabuka

Ne samo po imanu valika britanska firma *Big Apple* koja se do skara bavila uglavnom prodajem igrački i logičkih igara odlučila ja da se okuša i na kompjuterskom tržištu — izdvajen je milion funti koji će biti uložen u razvoj i dalja neimenovanog ogranka koji će lansirati 20—25 igara godišnja. Upravo je u toku otkup licenci za raznorazne arkadna igre zasnovana na poznatim filmovima.



Komarcijalni softver

Samo za prazentacije

Korlsnici PC računare laserskih štampače i programskog paketa *Ventura* koji se uz to bave nekom oblašću nauka nedavno su dobili još jedan dobar način da investiraju 75 dolara — američka firma *Specific Solutions* (telefon (415)941-3941) prodaja grupu fontova specijalno prilagođenih naučnim takatovima i prezentacijama. Ista firma prodaja i *FormEdit* specijalnu GEM aplikaciju koja obezbeđuje razna praprav ka ovih azbuka.

Inostrani život

Vreme i magija



Poznata britanska firma *Level 9* dvadesetorici srećnika nudi izvanrednu prilliku da zarade ovaj divni digitalni set i ovu divnu igru za atari ST odnosno amigu. Treba samo odgovoriti na dva laka pitanja (Koje je najčešće prezime namještenika firma Level 9 — nuda se prezimena Jones Smith i Austin — i Koliko ja seti u Njujorku kada je u Londonu 5 popodne) i da odgovor pošaljete na adresu *Popular Computing Weekly Greencoat House Francis Street London SW1P 1DG*.

Inostrani život

RENAME Activision Mediagenic



Firma *Activis on* po mnogima vodeći internacionalni distributer kompjuterskih igara donela je u poslovnom svetu prilično neuobičajenu odluku da na sebe primeni komandu *RENAME* — novo ime je *Mediagenic* što treba da asocira na razne CD ROM-ova i slična medija masovna memorija na kojima će *Mediagenic* zasnivati budući razvoj

Vredi pomenuti da je pomal firma *Activision* u proteklo fiskalnoj godini bio 45 3 miliona dolara što predstavlja povećanja od celih 70 odsto u odnosu na prethodnu godinu Velika stvar — svaka naša firma ja u toku ova godina povećala obrt za nekoliko stotina procenata i to prevashodno zahvaljujući neznatnoj inflaciji!

Strani časopisi

Časopis, disk i 500 funti



Prvi časopis koji je umesto na papiru izdavan na kaseti varovalno ja bio CLOAD koji je objavljivao programa za TRS-80 (čitaoci Galaksija 108 imali su priliku da ga upoznaju pre samo pet godina) Danas na papirni časopisi nisu ništa posebno atraktivno ali ponuda disketnog časopisa *Jumpdisk* zaslužuje pažnju — u junskom broju objavljen je šahovski program za amigu koji skromno tvrdi da će vas totalno uništiti i nudi čitavih 500 dolara čitaocu koji pobedi računar Žalite da sa praipadate? Adresa je *1493 Mr View Avenue Chico CA 95926 USA* a godišnja pretplata oko 70 dolara

Mreže

Samo za vlasnika „komodora"

U prošlim Računarima smo pisali o besplatnom modemu koji dobijaju pretplatnici *Compunet*-a koji žele da se zabavljaju multikorisničkom igrom *Federation II* Vesti smo prauzeli iz britanskog nedeljnika *Personal Computing Weekly* koji je nedavno objavio i dopunu — ponuda se odnosi samo na vlasnika Komodora 64 i 128 dok vlasnici celaih računara trabe da žrtvuju 135 funti Tak da komodorovci znaju koliko su srećni!

Računari u iziogu

Komodorov kolt



Koli više nija samo oružja kauboja — Commodora je upravo najavio PC kompatibilca koji će se zvati *Colt* i prodavali za samo 900 dolara Računar je zasnovan na mikroprocesoru 8086 koji radi na 7 16 MHz opremljan RAM-om od 640 K i dopunjan dvema disk jedinicama od 360 K Na znamo šta vi mislila ali jednostavno ne možemo da povarujamo da će Commodore uz ovakva cena prodati previše Kolloval

Komercijalni softver

Još malo kriptografije

PC računari koji stoje na raspolaganju brojnim korisnicima lakko mogu da predstavljaju čudo sigurnosti — svako može bez ikakvih problama čitati menjati i brisati tuđe datoteka! Francuska firma *Memorex Telex International* (telefon (1)47-80-14-00) tvrdi da je ailčnim problamima odzvonilo — programski paket *Eclipse* kriptografiše sadržaje proizvoljnih datoteka sa podacima korisnači algoritam jadinstven za svakog korisnika Uvadan je i suparusar koji može da čita i modifikuja svačija datoteka i koji dodaljuja lozinka cetalim korisnicima — baš kao na nakom VAX-ui Čilavo osiguranja košta oko 160 dolara

Komercijalni softver

Tekst procesor za Japanca

Dok za srećnika koji koriste latinicu prelazak na računarsku obradu lakala zahteva jedino kupovinu lakal procesora sirotl Japanci moraju da kupe kompletan računar sa spacijalnim akranom lasiaturom i softverom Sharpov pakal zvani WS-5501 obuhvata sva la komponante zajadno sa spacijalnim programom za kontrolu gramatika i spalovanja i košta oko milion jana Vredalo bi ga vidati!

Džepni računari

Snaga u džepu



Novi Csalov džepni podsatnik ja i poread malih dimanzija (128-74 mm) i niaka cena (100 funti) veoma mačna naprava — QWERTY tastatura LCD akran 16•8 znakova 32 K RAM-a i softver koji obezbeđuja pamćenja 1500 talefonskih brojeva i tarmina nakih 1000 sastanaka. Vradi pomenuti i ugrađeni časovnik realnog vremana dopunjan (ne baš) vačitim kalendarom

Operativni sistemi

Juniks za „amigu"

Unix se polako all sigurno približava raznim malim računarima — Commodore ja najavio *System V Unix v3.7* prilagođan i amigi 2000 koja ja opremljana akcaleratorskom karticom i hard diskom Akcaleratorska kartica ja da sa podsetimo zasnovana na mikroprocesoru Motorola 68030

Kada će sa junika pojaviti na tržištu? Pa. nešto posla akcaleratorska kartica!

Programski jezici

Za ljubitelje fotrana

Najstariji vrši programski jezik ja i poread najazda raznih C-eva i Modula liakako živ — komaija X3J3 amaričkog biroa za standarda ubrzano dovrševa specifikaciju nove varzija fotrana koja će sa po logici stvari varovalno zvati fortran 88 Programari su međutim nsaripljivi pa su dva člana komisija *(Michael Metcalf* i *John Raid)* nedavno objavila knjigu koja sa zove *Fortran 3X explained* — iako još na postoji ni tortran 6X standardni fortran 8X pravodilac knjiga ja veoma intaresantna jar prikazuja evoluciju r nsatanak jadnog novog jazika Od praktičnih atvari u knjizi čala naći labela raznih instrukcija i deklaracija koje će po svemu sudeći iziateti iz fortrana Da nabrojimo sama neke arilmalički IF BLOCK DATA izračunail GOTO DOUBLE PRECISION, DIMENSION COMMON i EQUIVALENCE Što sa prvih pet liča nečeme zapitakati All tortran bez COMMON i EQUIVALENCE nija lako zamisliti!

Komarcijalni softver

Sa CP/M-a na DOS

U moru programa koji obezbeđuju učitavanje CP/M diskete na PC računarima izdvaja se program DosDisk američke firme *PluPerfect Systems* (telefon (213)395-4664) koji nudi obnovu ustupa — pošto ga instalirate na bilo kojoj CP/M mašini (pa i našem TIM-u 011) ovaj program će bez problema učitavati i upisivati podatke u DOS formatu formatirane diskete je ne žaloti nemoguća Program će vam oduzeti 5 kilobajta i 30 dolara

Računari u izlogu

Gde je NeXT



Pažljivi čitaoci Računara se svakako sećaju da je Stiv Džobs *(Steve Jobs)* najavljivao svoj sledeći računar zvani NeXT za februar Vreme prolazi a od računara ni traga — priča se da su za zastojak krivi zastoji u razvoju prambiciozno zamišljenog softvera Najzanimljivija je međutim pitanja tržišta na koja je računar usmeran — multimilijarder *Ross Perot* koji je jedan od članova NeXT-ovog upravnog odbora je nedavno izjavio da će NeXT biti idealan računar za vladine i vojne potrebe Ako bi se međutim održala obećavana cena od 10 000 dolara NeXT bi imao sve šanse da postane poslovna mašina čiji će procesor (MC 68030) u mnogome prevazići razne PC-je i AT-e Postavlja se naravno i pitanje kakiko će zakasneli NeXT moći da se takmiči se drugim izvanrednim mašinama kao što je Sun 386 (ne slici koje su već čvrsto zauzele prostor na tržištu

Perifarijska oprama

Novi multisync monitori

NEC-ova linija multisinhronih monitora od nedavno je bogatija se dva modela *GS MultiSync* je 14-inčni monohram monitor koji po potrebi emulira kolor sfeke CGA EGA ili VGA kartice koristeći pri tome 64 nijanse sive boje Samo se po sebi razume da se monitor potpuno automatski prilagođava bilo kojoj od pomenutih kartica ili neizbežnom Herkulesu

MultiSync Plus je kolor monitor po mnogo čemu sličan slavnom prethodniku Dijagonala mu je povećana 15-inča dok je grafička razolucija sada 1024-768 *MultiSync Plus* podržava CGA EGA EGA Plus i VGA karticu i spi makintoš 2

Cena ova dva modela su 200 odnosno 950 funti a NEC-ov britanski telefon je (1)993-8111

Računari u izlogu

Još jedan džepni PC



Naprava se slike liči na sve osim na ono šta jesta — radi se varovali li na e džepnom PC-ju! Računar je zasnovan na CMOS varziji mikroprocesora 8086 opremljen RAM om od 128 kilobajta (proširuje se do 640 K) LCD ekranom 25•6 i minijaturnom QWERTY tastaturom koja je dopunjena numeričkim segmentom standardne veličina Dimenzija su 10*3 cm težine na prilazi pola kilograma a potrošnja je dovoljno mala da 4 AA baterija obezbeđuju trideseiak časova napreskidnog rada Računar je naravno opremljen disk jedinicom od 3 5 inča i RS-232 interfejsom koji obezbeđuje jednostavnu komunikaciju sa drugim PC računarima i uz dodatni modem kompjuterskim mrežama Cena ova džepna mašina budućnosti je 2000—2800 dolara a američki telefon proizvođača (617)663-7677

Računari u izlogu

Novi Atarijev PC

Posle nekoliko manje ili više (ne)uspešnih klonova Atari je odlučio da pokuša još jednom radi sa o AT kompatibilcu koji zajedno se 512 K RAM-a EGA/Herkulas karticom monitorom jednom disk jedinicom i pet slotova za ekspanziju košta 1300 funti Skupo? Nismo ni rekli da atari AT ' iraba kupovali!'

Perifarijska oprema

Makro tastatura

Na PC-ju se zovu BAT na VAX-u COM na SSC-ju EXEC Radi se ukratko o datotekama u koje upisujemo niz komandi i koje zatim startujemo kada iraba obavili neki veći posao dok mi štamo računar ima utisak da mu komanda neprekidno stiže sa tastatura Ovakve datoteke na žalost nisu svemoguće — česti su komarcijalni programi koji ignorišući operativni sistem direktno očitavaju tastaturu i koji prema tome ne mogu da koriste podatke iz komandnih fajlova Problem se na sraću može napisati i na drugi način — specijalizovanu PC i AT tastaturu firma *ECCO* (američki telefon (714)835—6000) opremljena su memorijom u koju slobodno upisujemo makro programa Komarcijalni program koji se direktno obraća tastaturi nama dakle nikakvog načina da sazna da li su podaci neposredno otkucani ili su ranije uskladišteni u memoriju!

Ova veoma zanimljiva i na baš previše skupa (150 dolara) tastatura bi naročito dobro došla raznim inatiluima i laboratorijama koje PC koriste kao terminal — dosadilo nam je da stalno kucamo KERMIT pa USERNAME

Domaća scena

Prvi domaći Tetrisi

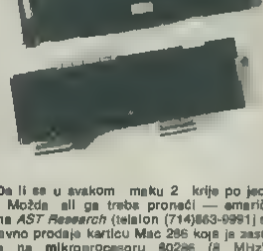
Sećate li se Tatrisa prva sovjetska kompjuteraka igra koja je mnoge ozbiljne programara iz raznih domaćih inatituta i laboratorija pretvorila u strašna igrače? Neki od tih igrača sada opet posteju ponovo programari lapiujući prve domaća verzija Tatrisa Zašto da pišeta „Tetris kada je već napisan? Zato što u laboratorijama nikada nema dovoljno PC-a a onaj višekorisnički VAX III Packard ili nako stalno zvrja prazni

Priča se ipak ne završava na programima se velike računara — našim čitaocima dobro poznati *Ivan Gerendir* prilagodio je „Tatris timu 011 što bi ako Redakcija doda do naka kopija moglo da znači da će sledeći Računari i razna druga BIG2-ova izdanja žastoko kasnih Glavni problam raznih programara koji implamantiraju Tetris' ja bodovanja koje je u originalnoj igri po svemu sudeći izuzetno složeno!

Pošto su potencijali Tatrisa uglavnom iscrpljeni (živ čovek na AT-u može da postigne nekih 10—12 hiljada poena dok se na 4 77 MHz PC-ju govori o nekakvih 25 000) hakarski gomji dom se dao u potragu za novom igračkom i pronašao trodimanzionalnu arkadnu avanturu sa kilometarskim imenom *Leisure Suit Larry in the Land of the Lounge Lizards* američka firma *Sierra* Reč je o igri sa odrasla" (pra nego što počnete morate da dokažete da ste odrasli rašavajući itekako kompliKovan test) u kojoj traba ubići sve barova kockarnice hotele i dragostore u gradu uraditi šta se već može uraditi i na kraju ataći u Penthausov spartman u kome vas očakujući nalzbežnu jabuku čaka izvesna Eva Tri saveta koja ća vas dalako dovesti (1) dijamantski prsten ja na umivaoniku WC-a (2) u baru traba uzati čašu viskija koju ćeta u susednoj sobi razmanili se daljnjski upravljač koji obezbeđuje teaplaan ulazak na jedno vrlo zanimljivo meste i (3) na bračno putovanje svakako traba poneti nož koji dakako, nećeta isprobati na svojoj dragoj Fawn već na konopcima kojma će vas dotična vesali pra nego što vas opljačka Još jedan dodatni savet? Kada presečete konopac ponesite komadići

Proširenja

AT u „meku"



Da li se u svakom meku 2 krije po jedan AT? Možda ali ga traba proneći — američka firma *AST Research* (telefon (714)863-9991) ednedavno prodaje karticu Mac 286 koja je zasnovana na mikroprocesoru 80286 (8 MHz) i depunjena RAM-om od 640 K i hardverom koji emulira CGA i Herkules grafički standard Poćte tatelu utaknete u bilo kaji Makintošov ekspanzioni slot, Apple-ov računar će progovoriti DOS učitavati DOS diskete i uopšte, postati na (baš sasvim kompatibilnij jEM PC AT Prvi testovi na žalost, nisu baš povoljni — mnogi MS DOS programi jednostavno ne rade dok je kad nekih drugih slika na ekranu prilično mutnjikavi

Programski jezici

Ze ljubitelje ekstenzije

I pored selidne standardizovanosti fortran nije imun na razne i medusobno ne baš kompatibilne ekstenzije Tretiranja tih ekstenzija postaje mnogo lakše pošto nabavite *Forcheck 8 0* programski paket nastao na psihološkom (!) odeljenju *Leiden* univerziteta (Holandija) Radi se u stvari o nekoj vrsti predprocesora koji će programa pisena uz korišćanja neka klase ekstenzije prilagoditi nekoj drugoj ksei ekstenzija III čak standardnom fortranu

Iako se *Forcheck 8 0* izvršava na PC kompatibilnim računarima njegove mogućnosti očito značajno prevazilaze tržište malih mašina! Cene od 325 dolara zvuči relativno razumno

Nova tehnologije

Portebl bez baterija

Zamislite računar koji se ne napaja ni iz gradske mreže ni iz baterija Perpetuum mobile? Na baš — zahvaljujući novom patentu škotovih laboratorija čiji su stručnjaci uspeli da svedu Tesiin transformator na dimanzija mikročipa I firma Sonic koja je taj patent primenila u prakei mali potrošači poput radio prijamnika i računera mogu da sa napajaju iz vazduha pretvarajući posiojaće RF signals u jednosmernu struju Neprava je veoma laka pouzdana i jeftina — dvadesetak dolara

Možemo da očekujamo da će ovaj pronalazak imati veliki impakt na našu zamiju — čim RTV ahvati da se signali koje pored ostalih i oni emituju mogu koristiti na razne načina poskupeće TV pretplate!

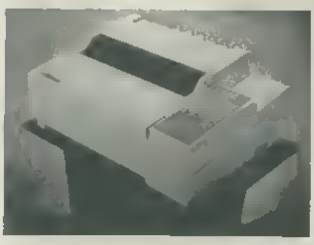
Inostrani život

Teško piretima!

Možda ste ponekad poželeli da umesto u ovoj našoj ekonomskoj krizi živite negde na Zapadu? Ni tamo nije lako — britanski vrhovni sud je nedavno potvrdio prasudu po kojoj će *Jeffrey Batty* vlasnik softverske bibliotake *Oman* morati da plati 4 400 funti i ostana bez kompletne oprema koja obuhvata i 3 000 piratovanih programa Ako jednom softvarske kuće napadnu naše pirate sudovi će biti pretrpani a blagajne prepune!

Štampači

Čudo od brzine



Ovaj simpatični matrični štampeč očito veoma specijalizovana firma *Output Technology* (američki telefon (800)468-8788) zeva se OTC 2161 isplauja neverovatnih 900 linija u minutu i košte samo 6450 dolara

Mreže

PS/2 u mreži



IBM-ove propuste drugi začas ispravljaju — veliki plavi još nije našao vremena da poveže svoje PS/2 računara u lokalnu mrežu I to je bilo sasvim dovoljno za firmu *Medge Networks* (američki telefon 20404-5851) I njanog vlasnika *Roberta Madgea* — u prodaji je sistem *Token Ring* koji omogućava povezivanja Micro Channel računera u mrežu izvanrednih karakteristika Najvijani su i dodaci koji će obezbedifi pristup i za obična XT odnosno AT kompatibilne računara

Inostrani život

Život jednog virusa

Po žiu poznati virus za smigu zvani *Byte Bandit* (pojedajte prošia Računara) upravo je doiljao — uništavaju ga programi *Random Access Virus Killer 2 1* i *Tristar Virus Killer* ovaj poslednji „antibiotik" (prodaje ga firma *Hi Voltage Computers)* ja po svamu sudeći najbolji jer uništava i viruse sa kodiranim imenima SCA i CCW Odgovarajući antibiotik za PC kompatibilne računara I njihove viruse zvane *Brain* i *Lehigh* je *PC Immunise* Kada bi i virusi raka AIDS-a i drugih boleština mogli da se utamane ovako lako i brzo

Inostrani život

A cene pedaju. .

Bolje bi bilo da je ove vest dobila podnaslov „Odmaća sosna ali — bolja išta nago ništa Commodora je smanjio osnu svog portebl PC računara PC1 sa 500 na 370 funti Najavljano je i pojeftinjanje ostalih PC računera ali se o jeftinijem C64 i dalja na govori — izgleda da je osne već probila sve plafone!

Inostrani život

Sinklerov sledeći korak

Konfraverani Klajv Sinklar uporno odbija da razgovara o novim računarima koje razvija njegova nova kompanija (priča se da će se računari pojaviti tek sledeće godine) ali se zato ne sva usta raspričao o biciklima — zar niste eduvek želali bicikl koji kako Sinklar kaže kerlafte a da to i ne primećujete?

Inostrani život

Korist od slabog dolera

Tandiju očito ne smete pad dolara — šel prodaje ove velika japaneke kompanija *Richard Risser* tvrdi da je smanjanje vrednosti američke velute omogućilo nedavno sniževanje cena koje zaviano od modela iznosi 50—100 funti Veoma popularni PC kompatibilni model 1000 SX zajedno sa dve disk jedinice hard diskom od 20 M i monitorom seda košta 1200 funti Tandijev britanski telefon je (922)710-000

Komarcijalni softver

Borlandov „Sprint"



Dugo najavljivani takst procesor *Sprint* firma *Borland International* pojaviće se u julu a na u septembru Izjavio je Borladov direktor Filip Kan *(Philippe Kahn)* Cena još nisu objavljana ali će *Sprint* verovetno koštati oko 300 funti

Fotografiju Filipa Kana dopunjavamo fotografijom Turbomana novog junaka koji reklamira Borlandove programske jezike

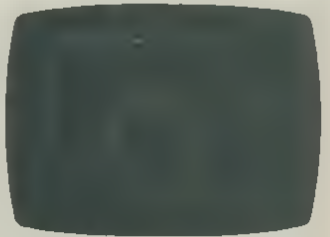
Inostrani život

Pit i Pem zeraduju

Pre nekih osam godina mladi britanski bračni par je obnovio kompaniju koja će sa baviti uvozom softvera za apl Mala kompanija je postepano širila ecsao tako da je tokom prošla flakalne godina ostvarila obrt od 70 miliona funti i real profita od 105 posto ima ja međutim i dalja P&P po inicijalima vlasnika

Komercijalni softver

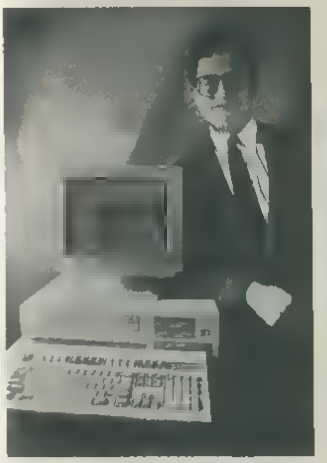
Slike bez granica



Na veoma dinamičnom tržištu gralika okušala se i firma *Manufacturing Services* (američki telefon (714)951-6858] Programski paket Anvil-500pc predstavlja kombinaciju programa za slobodno crtanja CAD paketa i tekst procesore kompatibilnog se EGA i VGA standardima Ukupne osne dostiže 6000 dolara

Računari u izlogu

Bitka na Mikro kanalu



Internacionalni konkurs zvani prvi PS/2 klon očito se rasplamsava do usijanja je prvi stigao *Michael Dell* (na slici) koji je upravo promovisao svoje modele 400 i 500 potpuna klonove PS/2 80 i PS/2 60 leko će računari biti u prodaji tak početkom jeseni kompijuterski novinari su datelje testirali prototipove i utvrdili da je poneSanja VGA kartice i veome kritičnog disk kontrolora prikačenog na jedan od portova kloniranog Mikro kanala završeno Cena sa zavisno od konfiguracija kreću izmedu 2700 i 3600 funti

Olivetti je u međuvremenu odustao od kupovine licence ali ne i od Mikro kanala — za kraj godina su najavljene prve PS/2 kompatibilne mašina ali se još ne zna da li će one biti rezultat dogovora sa IBM-om ili neklm drugim proizvođačem

A šta ne to kaže IBM? „Veliki plavi" je zvanično izjavio da sa neće protivili proizvodnji PS/2 klonova uz uslov da se specijalno dizajnirani čipovi nabavljaju od nekog ovlašćenog proizvođača ili od proizvođače koji neme nikakve (pa ni špijunske) veze sa IBM-om — prihvataju se kopije nestele samostalnim razvojem ali na i do skore prilično uobičajena krada dizajna nekog čipa!

Hardver

Monitor majstor ST

Ako imate oba monitore (mono/koler) za ST-a znate proceduru oko zamene — isključiti kompjutar izvući konektor prikljućiti drugi uključiti kompjuter Ali sada sa *Monitor Master-om* firme *Practical Solutions* ova što treba da uradite je da pritisnete prekidač i — gotovo Sistem se sastoji iz kutije u kojoj su smešteni portovi i prekidač i kabla koji se spaja sa ST-ovim portom za monitor Pored toga postoja i dva RCA konektora koji daju kompozitni video signal i audio signal Na žalost da bi ove funkcionisalo potrebun vam je RF modulator (512 STFM) Upozorenje! Nemajte nikad držati monitore jedan pored drugog (zbog različitih frekvencija) već isključivo na odstojanju od najmanja 40 cm

Mnogi od vas će misliti da je 50 dolara koliko košta evaj uređaj mnogo para za jeden praktičnik dok ne uvide kolike je cene opravka kablova i konektora koji su bili previše u upotrebi Telefon je 602-864-9612 USA.

Stono izdavaštvo

Nova verzija „Publishing Partnera"

Soft Logik je izbacio novu verziju *Publishing partnera* pod imenom *Publishing Partner Professional* U odnosu na staru ova verzija ima puno novih opcija uključujući i automatsko namaštenje takste oko grafike nepravilnog oblika rečnik za proveru spelovanja, grupisanje objekata učitavanje slika u *DEGAS* kompresovanom *Easy Draw* ili *IMG* formatu Undo brže štampanje sigurnosno zaključavanja objekata slobodne rotirenje slike t tekste slobodno zumiranje šest novih fontova rad na 6 dokumenata istovremeno i precizno odredivanje veličina slove u inčima cm i mm sa preciznošću od t/3600 inča *Pablishing Partner Professional* kosta 199 95 USD

Proširenja

Proširenje mamorije ze „atari ST"

EZRAM II je novi proizvod Terrific Pheripherilals-a koji omogućava visanicime ST-a de povećaju memoriju računare de 2.5 M *EZRAM II* se prikljućuje piggyback matodom na Video Shifter i dobija se bez RAM čipova Cene je 139 95 USD (Terrific Pharipherals 17 Si Mary s Court Brookline MA 02416)

Progremski jazici

Raal Basic ne radi na „mege 2/4"

Ako imate Mega 2/4 i interesuje vas *Real Basic* Computer Crossware Labs-a načete moći da ga koristite U CCL-u još uvek nisu dobili Mega računare i niko nije mogao da proveri softvereku kompatibilnost

Komercijalni softvar

Intrecorp Inc.

Intrecorp nova ameriška softverska kompanija izdala je nedavno veliki broj neobičnih programa za „atari ST *Video Wizard* je kobinacija programa za štampanje natepnica za video kasete titlovanja filmova i baze podataka *Bumper Slticker Maker* i *Business Card Meker* vam omogućavaju da štampate svoje natepnice i vizit-karta *Love Quest* i *Interaction* služe za učvršćivanje vaših ličnih veza i poboljšanje ljubavnog života Cene su im 59 USD svaki Intracorp Inc 14160 S W 139th Court Miami Fl 33186

Igre

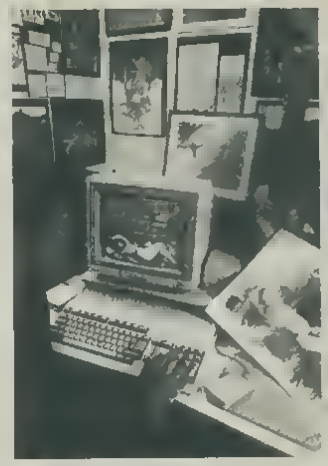
Prve saulska igra



Predstojeća Olimpijade u Seulu konačno je privukle pažnju softverskih kuća firma *Ocean* vas poziva da postanete prvi čovek u istoriji koji je osvojio zlatne medalje u sve tri olimpijska višeboja Treba vam samo „spektrum komodor 84 Amstrad i Atari ST „amiga III PC i program *Daley Thompson 88*

Računari u izlogu

Još jaftinija „amiga"



Pre mesec dana govorili smo o padu cena „amige 1000 a sada je slična sudbina zadesila i noviji model 500 cene je smanjene za 100 funti pa amiga 500 košte 400 funti! Obećano je uz to da će do kraja godine biti pokrenuta velika propagendna kampanja i da će biti lansirani brojni periferijski uređaji Ovo pojeftinjenje je veliki poen za Commodore čiji je veliki konkurent Atari nedavno ćak povećao cene ST računara!

Nezvanično

Tramial&Jobs Inc.?

Da li sta čuli glasine o tajnom ugovoru izmedu Ataria i kompanije Stivena Dž0bsa *NeXT Inc?* U Americi kruža priče da su sa ove dve kompanije udružile i da planiraju proizvodnju jaftinih mekintoš klonove Moguće je da Džobs sklopa tajne ugovore ali na sa Atarijem Uostalom atari ST već može da koristi mekove programa pomoću *Magic Sac-a* i *Aledina*

Hardver

Brain STorm

Atarijevci! Eva jedinatvena prilika da za malo para nabavite sve o čemu ste do sada mogli same da maštate Hard Disk 30 Mb 3 5 inčni drejv 72K 5 25 inčni IBM drejv 360 K bataterijski časovnik i regulator napona sva u tstoj kutiji sa ugrađenim lenom za hladenje *Brainstorm* kake sa ova čudesne naprava zove košteće vas samo 849 95 USDI (Brainstorm 911 E Pika Sulte 325, Seattle Washington 98112)

Komercijalni softver

VIVA!

VIVA (*Video Interactive Visual Authoring*) grelički orljentisan jazik za pravljenje prezentacione grafike omogućava izradu animirane prezentacije sa mešanjem tekata grafika muzika i govore na vašem atariju ST *VIVA* ima preko 200 grafičkih procedura i 50 video komandi i može sa priključi na monitor III video rikorder (*VIVA* — $249 Konowledgeware 576 Talegraph Pontiac MI 48053)

Igre

Rembo ponovo napeda



Paralelno sa novim Stalonevim hitom Rembo III koji upravo puni blagajne američkih bioskope (ima li neki beogradski video klub kopiju?) *Ocean* ubrzano priprema istoimenu kompjuteraku igru koja će biti prilagođena „spektrumu komodoru 64 amstradu i stariju ST „Amigi i PC-ju Razni britanski humoristi su reagovali „na prvu loptu — na karikaturi piše da su sa „Sovjeti povukli iz Avganistana u poslednjem trenutku — Rambo samo što nije stigao!

Komercijalni softver

LDW bejzik ze ST

Firma *Logical Design Works* (LDW) izdala je novu (2.0) verziju svog LDW bejzik kompajlera Paket obuhvata više od 50 novih naredbi koja pristupaju GEM-u mogućnosi editovanja kompajliranja i izvršavanja programa bez napuštanja LDW bejzika(1) povećanu brzinu aritičke i dinamička pokazatelja i multi-linijske funkcija i procedura sa parametrima i lokalnim varijablama Cena od 80 dolara uveliko premašuja cenu od 25 dolara koliko je koštala prva verzija Američki telefon je 408-435-1445

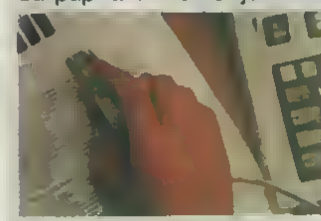
Periferijska oprema

ST džojstik ze simulacije letenja

Pasle pojave sličnih, i *Microflyte* je izbacio na tržište džojstik za *Flight simulator* program Priključuja se na port za miša i omogućuja 3 moda upravljanja standardan stari džojstik proporcionalnu kontrolu i promenljivo-osetljivu kontrolu koja može da se podešava Kutija sadrži palicu 4 prekidača i lampicu (on/off) Cena je 120 dolara za detalja a možete zvati 703-777-7157 USA

Periferijska oprema

Sa papira u memoriju



Ako već kupite OTC 21P1, potreban vam je neki način da u svakom minutu nabavite 900 linija koje će štampač ispisati Dobar početak je optički čitač DataSweep firma *Soricon* (američki telefon (800)541-7268) — tvrdi se da je tačnost 99 3 procenta a brzina skeniranja 120 reči u minutu pri čemu su ovakvi podaci uvak podložni sumnji Šta se cene liče najizvesnosti nama — DataSweep košta 1250 dolara

Komarcijalni softver

Super digitalizovenje

Trio programara koji su napisali *Spectrum 512* program za starija ST koji prikazuje 512 boja u isto vrama pojavio sa sa novim proizvodom *Digispec* omogućava digitalizovanja slika u punih 512 boja sa *Computer Eyes* digitalizatoram Ponad ST-ova palate od 512 boja *Digispec* koristi šara sa dva ili više boja da simulira preko 24000 boja! Program čita i amigine IFF slike i prikazuja ih u svoje programu

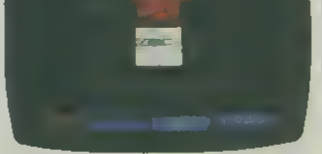
Hardvar

Megadiek

Prijatno iznenađenje nam dolsai od Firma *Alpha Systema* proizvala ja uradaj zvan *Megadisk* On je 500% brži od hard diska omogućuja učitavanja bilo kog programa I ima ugrađen baterijski napon u slučaju nestanka struja ili isključenja računara *Megadisk* ja u stvari 1 M RAM disk koji sa jednostavno priključuja na ST-ov kartridž port Ima I opciju za odvajanja 200 K za printar bafer što je veoma korisno jer sada možete raditi druga stvari sa računarom dok štampač radi Uređaj sa može ponašati kao poseban drajv ili ga možete pomoću prekidača nadograditi na postojeći RAMI Postoji i varzija sa pola mega ali sa zove *Polydisk* Baterija su izmenjiva i traju oko 3 godine *Megadisk* košta 300 a *Polydisk* 200 dolara Američki telefon 216-374-7489 od 0—24 h

Komercijalni softver

Industrijski CAD/CAM



Računari sa ne primenjuju samo u čistim i lapo ostvatljanim radnim sobama — surovi fabrički uslovi su upravo izvanredne prilika da sa testira robusnost računara i kvalitet softvera Što sa računara liče brojne firma po upledu na IBM proizvoda industrijska varzija svojih PC kompatibilaca A što se tiče softvara vredi sa obratiti firmi *Universal Automation Inc* (američki telefon (603)880-6653) Najnoviji hit program sa zove *FloPro 2.2* i obezbeđuja kontrolu raznih uređaja pri čemu oiklus ponavljanja jednog objekta iznosi 5 milisekundi

Programi sa na zadaju ni na bejziku ni na paskalu pa čak ni na asamblaru — programer interaktivno unosi blok šemu delovanja kontrolera a onda sa ta blok šema izvršava poput svakog programa U toku razvoja blok shema se mogu testiroli putem simulacija što bitno smanjuja opasnosi od požara koji bi nastao samo zato što ja programar zaboravio neki IF Samo se po sebi razuma da *FloPro* radi u višeprogramskom režimu

Hardverski minimum za izvršavanja programa ja PC sa 512 K RAM-a EGA karticom i 132-kolonskim štampačem A finansijski minimum? 800 dolara

Nonsense in Basic

Neki mračni virusi

Zaprepašćuje me činjanica da su avi podmatači virusa naki neinventivni, mračni, destruktivni i jednostreni tipovi Na umaju ništa drugo da smisla nego da ti obrišu hard disk ili pokvare program Da je bar neko od njih lole ter, uz virus bi uneo i upozoravajuću poruku Zašto da dzentimanski ne pruži šansu? Recimo

„Virus je u tvom računaru već dva meseca Ako ga za jedan sat ne otkriješ i uništiš, oprosti sa od svega na hardu

Da nije te destrukcije i jednostranosti, hakerski život bi bio mnogo zanimljiviji Vlasnici računara bi mogli divno da se zabavljaju čitajući poruke izazvane bezazlenim virusima čiji su tvorci ljudi različitih-karaktera i akcionasti

Romantičan podmatač virusa bi verovatno razasilao ovakvu poruku

„Hakerko mila, ma gde bila, zar na žudiš za nodima obasjanim titravom avetlošću monitora u kojima bismo do zora zajedno ukucavali šesnaestobitna registre?

Posiovan podmatač bi imao priliku za zaradu

„Ovaj virus je reklama za virus koji uništava virusa Garantovana dekontaminacija vašeg operativnog sistema

Naivan bi bio u stanju da uputi i ovakvu poruku

„Menjam spektrum 16K za IBM kompatibilac Šifra „Trampe bez dopiste

Politizovan bi se založio za akciju

„Mi treba da sagledamo eve socijalno-psihološke aspekte neopravdanog podmatanja virusa, a zatim moramo energično preduzeti adekvetne korake za daiji pomak u devirusizaciji

Sve u svemu s malo više maste virusi bi mogli da otvore neslućena staze komuniciranja Šteta što su podmatači takvi kakvi jesu Žive u anonimnosti teroristička gerila umasto da postanu hakersko organizovana snaga korisna za razvoj računarstva I na samo računarstva Ničim neoputana razmana ideja o obrazovanju, proizvodnji ljubavi produktivnosti vaspitanju, miru, razoružanju, politici ekologiji religiji grupnom seksu štetnosti pušenja umetnosti . doživela bi snažan razmah

Naravno, bilo bi i negativnih posledica Uzmimo za primar samo ovu poslednju oblast umatnost Nikakav recenzent niti kritičar ne bi mogao sprečiti placa tekatova za novokomponovanu muziku da široj javnosti putam virusa prezentira avoje poslednja atihove

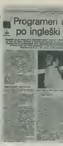
Otišao si sarmu probo nisi, ne može me utešit ni PiSi

Ali, stvarnost ne zavisi od naših želja. Virusi će i dalje biti isti, stereotipni i opasni

Bata Bajt

Otvoreno pismo Božidaru Travici

Nonsense balkanskog špijuna



Programeri uz more po ingleški govore

Svako ko prati redovno dnevna novina nedeljnika i ostala časopiza odavno za privikao na široki spektar zavičajnih klupskih i ostalih boja koja dominiraju u komentarima polemikama izveštajima Ovakav kolorit još uvek kod mnogih izaziva uzbuđenja što je razumljivo ali kod običnog čoveka takvi napisi sve rade izazivaju iznenađenja bez obzira da li je prijatno ili neprijatno

U broju 40 RAČUNARA jedno takvo iznenađenje nam je priredio Božidar Travica posetilac računarskih zajmova i vašara svojim izveštajem sa YU zajma softvera održanom u Splitu Zapravo taj izveštaj na iznenađuje u celini već samo onim delom koji više liči na omušenu dostavu podozrivog balkanskog špijuna nego na plod medijativnog razmišljanja potencijalnog novinarskog gurua Izgleda da je ovog puta podozrivi duh uvek spreman za borbu protiv mrakih mu pirata softverskih nadvladao korektnost izveštača a to je ono što iznenađuje bar kada se radi o napisima u časopisu za popularizaciju informatike i računarstva

O čemu za zapravo radi U spomenutom izveštaju za sajma softvera objavljenom pod duboko osmišljenim pzaničkim naslovom Programeri uz more po ingleški govore B T je pored zalalog obradio i udarnu temu o CASE alatima i piratima i/j engleskim pličicama Centralno mzato dobila je lakra Delta za svojim generatorom aplikacija AGP a kao uspul ili možda na sama usput hrabra i principijelno B T je denuncirao njemu znanog a čitaocima neidentifikovanog visenika beogradska firma koja promoviše igra i uslužna programa Kao čovek uopće na kraju čime se bavi i još je naprotiv spreman (!?) da zapeva Pirat zam — tim za dičim

No vratimo se udarnoj temi case — lakra Delta Mučio je tih majsko-junačkih dana u Splitu našeg podozrivog B T i sajamski time sharing i Mara iz one turistička agencija i ko zna što još ali ponajviše „ingleški i AGP Na mah je posumnjao vrli borac protiv engleskih plićica u programersko pošlanja i tu je kako napise sumnji podlegao U telikoj meri da se prepustio ubadivanju lakra Deltinog „demonstratora a prilom ga oslo vrama mučilo i nije napuštala pitanja i postavio ga je u pogodnom trenutku i u stilu balkanskog špijuna vešta amelo i sugzativno Zašto su u AGP-u sve maska poruka i komanda na engleskom jeziku?! dobio je odgovor kakav za i može dobiti na takvo pitanje i detaljno objašnjenja da korisnici AGP-a traža isključivo anglzaku verziju Dovoljno da se razbija sumnja? Ma na obrnuto sumnja u pošlanja programersko se povećala! Kakve veza imaju s tim korisnici i njihova jezička preferencija i to baš ingleškog ? Naš borac protiv pirata postavlja razarajuća pitanja o poreklu programera i to onoj koja dolazi iz izvora Zadovoljan odgovorom Opal na! Za to prelazi na završni udarac je profivnik je na kolenima Božidar Travica sa predstavlja ke je i šta na prezentaciji traži što je verovatno trebalo da impresionira prisutne i demonstratora i postavlja završno sugestivno pitanje A cena nije u dolarima? Tačno Napokon! Dokazani postupak je završen jer cena je u dinarima A zaključak? Iscrpljujuća dvosatna koncentracija nema ploda Na pitanja više ni zaba jerbo kako reče pitanje jzano ipak lucidno zaključuje da bi bio u koži prodavaoca softvera da bar maska na jezik očeva svojih Sveka čast

Božidare Božidara šta Vam je to trebalo? Šta Vam je trebalo da se bavite takvim prizemnim podmetanjem i insinuacijama Vaša borba protiv piratizma je pravedna ali prilaup dilatantski naprimarena Verovatno Vam je ustupio prostor u dnevnom kriznom vremenu Da ste pitali saznali bi da su AGP razvijali stručnjaci za softver iz četiri republika Srbija Makedonija Hrvatska i Slovenija i da su inspiraciju tražili u lapu napisanih funkcionalnih zahtijeva 4 generacija a ne u borbi za čistoću ravnosti maternjeg jezika protiv najezda maternjeg im engleskog Pitajte Božidara ako ne znate kako je teško razvijati i održavati i prodavati takav softver kao što je lakra Deltin AGP Koji trebe pored funkcionalnosti i dobrih performansi da bude prilagođen za korisnika i to ne samo u Jugoslaviji bogatoj raznovrsnim jezicima neko i izvan naših nego i na stranom tržištu ili možda sumnjate i u to da postoje ljudi u ovom našem jugoslavenskom prostoru koji veruju da su u stanju samostalno da razvijaju i prodaju softver na stranom tržištu

I na kraju čemu ili kome bi trebalo da posluži omalovažavajući ton iskra Deltinog softveraša nazivate demonstratorom sa nasumnjivim trgovačkim talentom Programera ili to iz Vas infariorni stvaralački polenciji ili neki drugi naraščićeni računi? Zar je teško bilo prepoznati da je u Stanko Vuné mladi inovativni entuzijasta koji je pored rada na razvoju AGP-a prihvatio i teškog zadatka da proizvod u čijem razvoju učestvuje prezentira informatičkoj javnosti?

Do sladećeg zajma primite srdačne pozdrave

Vladislav Pintar

Najprirodnije bi bilo kupiti neku 80386 karticu sa nešto RAM-a i utaći je u neki od 16-bitnih ekspanzionih slotova — stari 80286 i dalje radi kao ulazno/izlazni processor RAM na staroj ploči je i dalje u opticaju koriste se stari hard i flopi diskovi i ostala kartice kompatibilnost za softvere je stoprocentna (sve i da neki program ne radi, 80386 karticu možete da isključite!) Činjenica je međutim, da ste na ovaj način nabavili tek pola 386-računara RAM je i dalje 16-bitno organizovan DMA kanali su 16-bitni ili čak 8-bitni komunikacija za diskom je čak sporija nego što je bila na AT-u, dve procesora gube nekakvo vreme za komunikaciju

Od koprocesorske ...

Zato je u poslednje vreme sve modernija kupovina kompletnih 80386 ploča koja zamenjuju osnovnu AT ili baby AT ploču ali zato komuniciraju sa starim karticama starim hard diskom i starom periferijom Ako za dobre karakteristike ovih ploča dopune činjenicom da je originalnu AT ploču mnogo lakše prodati nego ceo AT (teško je naći čoveka raspoloženog da odjednom plati nekih 500 starih miliona za računar dok ljudi koji bi kupili osnovnu ploču a zatim polako prikupljali ostale komponente sistema znamo bar nekoliko) čoveka zaista počne da svrbi dzani dlan (za neupućene sledi izvod iz „večitog kalendara *kad vas svrbi levi dlan primićete para kad vas svrbi desni daćete para*).

U ovim „Računarima ukratko opisujemo četiri 80386 ploče sličnih karakteristika koja koštaju između 1500 i 3000 dolara **Fortron 386** firme *Fortron, 2380 Qume Dr Suite F San Jose CA 95131 U.S.A.* **Micronics 386** firme *Micronics Computers 935 Benecia Ave Sunnyvale CA 94086 U.S.A* **Turnpoint 386** firme Turnpoint America 180 North Center St Suite 224 P.O Box 41334 Reno NV 89504 U.S.A. i **Mylex 386** firme *Whole Earth Electronics 1321 67th St Emeryville CA 94608 U.S.A.* Veličina svake od ovih tabli je 12·13 inča (30.5·33 cm) što grubo odgovara baby AT ploči i staje u standardno ili baby AT kućište U centru svake od ploča nalazi za intelov mikroprocesor 80386 koji radi na 16 odnosno *(Micronics)* 20 MHz podnožje za aritmetički koprocesor 80387 Koji radi na 16 odnosno 20 MHz ili stari 80287 na 8—10 MHz bar megabajt brzog (120 ns) 32-bitno organizovanog RAM-a specijalan BIOS ROM i potreban broj 32-bitnih 16-bitnih i 8-bitnih ekspanzionih slotova Budimo ipak malo precizniji i opišimo karakteristike svake ploče

... do matične ploča

Fortron 386 košta 2250 dolara i u tu cenu su uračunata 2 megabajta RAM-a na specijalnoj kartici koja zauzima jedan od tri 32-bitna ekspanziona slota (tu su naravno i četiri AT i dva XT kompatibilna mesta za proširenja Kartica, na žalost ne sadrži ni malo keš memorije što je čini potencijalno najsporijom od pomenutih 80386 tabli za trenutak ćemo videti da brzinski testovi potvrđuju ovu nesumnjivu teorijsku tetinu

U cenu *Micronics 386* kartice (2950 dolara) uračunata su dva megabajta RAM-a koja su zalemljena za osnovnu ploču — svih osam ekspanzionih slotova (tridesetdvobitni pet AT i dva XT) je slobodno Predviđen je i specijalni poklon za zaljubljenika u Tetris — preklopnik pomoću koga biramo radnu frekvenciju od 4.77 8 16 ili 20 MHz. U računar je ugrađen BIOS

Ploče 80386

Gerovitei ze AT-A

Četiri asa

Iako je AT kion prava premija za mnoge domaće korisnika računara, u prirodi kompjutariata je da na budu zadovoljni postojećim — zar AT sa njegovom 16-bitnom memorijom i 64-kilobajtnim limitom na bi bilo lapo zamaniti nekom mašinom zasnovanom na Intelovom 80386? Naravno da bi, samo to košta: bar 5000 dolara za kompletnu konfiguraciju. Stari AT vam u tom slučaju ostaje kao dobrodošla backup mašina (čovek nikada na zna kada će sa šta pokvaritil), naravno pod pretpostavkom da možete da priuštite backup mašinu od 3000 maraka. Ako ne možete a ipak želite da predate na procesor nova generacije, preostaje vam samo nako od proširenja!



koji semi birate *(Phoenix* III *Avard ROM.* U računar je ugraden BIOS koji sami birate *(Phoenix* III *Avard ROM* preporučujemo *Phoenix)* kao i rutine za relociranje EGA odnceno VGA BIOS-a u RAM koji se odeale mnogo brže od ROM-a III EPROM-a

Turnpoint 386 zajedno za dva megabajta RAM-a košta 2600 dolara u cenu su uračunata 4 kilobajta keš memorije koja ubrzava radi sedmokanalni DMA kontroler koji je karakterističan za većinu 80386 tabli CMOS RAM časovnika realnog vremena ja neuobičajeno dobro obezbeden što znači da sistem može na zamo da dataktuje nego i da ispravi dve greške nastale zbog nekog defekta na baterij!

Mytex 386 ja po svemu prilično kiselčna tabla — košta 1600 dolara uključuja megabajt RAM-a koji se na samoj tabli na može proširivati ako želite četiri megabajta, morate da kupite drugu verziju table koja košta oko 3100 dolara. Keš memorija od 32 kilobajta ja odgovorna za pristojna razultate na brzinskim testovima dok šest AT i dve XT slota obezbeduju standsrdnu ekapanziju Začuđujuća ja napostojanja 32 bitnih slotova za akapanziju — jeste da na tržištu nama mnogo kartica kojima bi ovi slotovi bili neophodni ali bi kupovina 386 mašina trebalo da predstavlja nekakvu investiciju u budućnosti

Što se softvraka kompatibilnosti tiče sve su se table pokazale izvanredno što znači da ni jedan od programskih pakala nija pravio nikakve problema u MS DOS modu Naka starija igre su naravno zahtevala usporavanja kloka lil su bila tolika brze da se ni tada nteu mogla igrati ali pretpostavljamo da ja odricanja od ovih igara zaavim prihvatljiv kompromis pri nabavci naka 386 mašina Što se OS/2 tiče ave table obezbeduju njegovu akapleataciju ali ja *Mytex 386* neophodno proširiti do četiri megabajta

Keko se ko snađe

Tabela 1 prikazuja rezultate Bajtovih benčmark testova koji su izvršeni u najboljoj mogućoj konfiguracijl svaka od tabli — aritmatički koprocesor maksimalan klok i tako dalja *Micronics* ploča ja na brzinskim testovima postigla najbalje rezultate što i nije tako neobično kada se uzma u obzir klok od 20 MHz i činjanica da ja RAM statiški (100ns) a na dinamički *Turnpoint* tabla ja naznatno brža od *Mytex 386* što je unakoliko neobično kada se uzmu u obzir kapaciteti keš memorija *Fortren 386* ja, kao što smo rakli brzinski šampion u sporosti

Ostalo je još da pomenemo dokumantaciju koja teško može da zasluži prelaznu ocenu Bez izuzetka svaka od pomenutih kartica je dopunjena uputstvom koja je previše lakonsko da bi odgovaralo početnicima i previše kratko da bi odgovaralo stručnjacima — savršeno je opisano gde koji konaktor traba prikljućiti ali nedostaju bilo kakve šema koja bi pomogla u trenucima kada nešto pođe po zlu Vredi naravna reći de je ovakva dokumantecija manja smatnja za Amerikance kojima su u svakom trenutku na raspolaganju telefonaki seveti proizvođača ali sa činjenica da dokumanacija 386 ploča preživljave dačja bolesti nekadašnjih uputstave za matrične štampače teško može nazvati progreaom!

Ugradnja svaka od kartica na predstavlja poseban problem — treba samo povazati PSU utaknuti disk kontrolar i video karticu i ava će biti u redu, to je teorija Što se prakse tiče mi se već dobar mesec dana mučimo se jednom 386 tablom (firma je *Plantron* što znači da se na radi ni o jednoj od pločica koja pominjamo) koja konstantno odbija da radi onako kako bi trebalo Ne želimo naravno, da tvrdimo da se na radi o uticaju Marfijevih zakona na poluispravnu ploču ali vas moramo upozoriti da 386 tabla koju u radnji nikako ne možete testirati predstavlja mačku u džaku šak i ako se isperušuje u plastične] kesi)

Priredio: Dejan Ristanović

Hewlett Packard Laserjet Series II

Laserski džet-set

Protekio je više od pola godine od kako smo objavili preliminarni test lasarskog štampača „Laser Jet Series II" Firme Hewlett Packard. U međuvremenu je na ovom štampaču složeno i prelomljeno nekoliko knjiga, uključujući i neke umetke „DOS 3.30", „Veuture" i „Turbo Paskal 4.0", i podosta rukopisa. Izvanredna iskustva koja smo stekli u primeni i eksploataciji ovog uređaja podstakla su nas da „Laser Jet" još jednom preporučimo pažnji naših čitalaca.

Štampač Hawlet Packard Laser Jet Series II je kao i veliki drugi zasnovan na novoj tehnologiji firme Canon koju odlikuju smanjane dimenzije visok stepen pouzdanosti i izuzetna brzina isporučuje se u paketu u koma se odvojeno nalazi kasela sa tonarom centronika kabl za povezivanja sa računaram i dva priručnika Da bi se štampač doveo u operativno stanja dovoljno je postaviti lonaraku kasetu u mašinu

Umetanje kasete

Povlačenjem dugmata na gornjoj površini oslobada se masivan poklopac koji se zatim može podići otvarajući put u unutrašnjosti štampača S obzirom da se u kaseti nalazi tonaru obliku praha pra postavljanja u štampač preporučuja se blago okretanje oko uzdužne osa da bi se toner ravnomarno raspomdio Kasala se postavlja u poklopac (naispnica sa oznakama za čuvanje kasete okranuta na gore) pa tako se ne može postaviti pograšno čak i ispravan položaj zbog dimanzija kazala zahteva prilično pažnje U svakom slučaju kada se pogodi pravao kasela ulazi veoma lako

Sladeći korak je oslobadanja pristupe tonera ka selanskom bubnju koji sa nalazi u samoj kaseti Sa desne strane poklopca nalazi se otvor kroz koji sa može videti kasela i plastična ploščica zalapljana za nju Pločicu traba allom odlepiti i zatim koristeći ja kao dršku izvući van kasete iraku koju u akladišnim uslovima spračava izlivanja tonera Ovaj postupak nija ni malo nažan — zahteva sa prilično jako povlačenja pa kod prve kasela može izazvati i malo straha od posledica Imali amo prilika da ovo ponovimo na nakolko kasela i uvek ja ava tekla na lali način tako da se očigledno radi o pradvidenoj proceduri Nija loše pri vadanju iraka drugom rukom pridržavali poklopac štampača jer imamo utisak da on ipak nija predviden za ovakva naprazanja

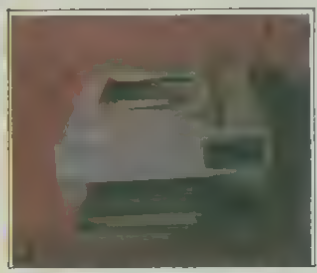
Jednom postavljanu kasetu na bi trabalo vaditi iz uradaja Ako ja to ipak neophodno iraba raditi bez naglih pokrata Uz svaku kasetu sa dobija i tilcani čistač larmičkog valjka Valjak koji sa zagrava i peča toner nansaan na hartiju sakriven ja lapod zajanog poklopca u zadnjem delu unutrašnjosti Čistač sa lako postavlja

Pogled u unutrašnjost štampača na otkriva ništa posabna interesanino — ava alektronika ja skrivena na dnu i uz bočne ivisa tako da je jadino vidljiv transportni mahanizam koji proviači papir i tanka žica (CORONA WIRE) koja je obezbedana od dodira relkom mrežom Pristup unutrašnjosti štampača je polrebaan samo u dva slučaja — pri zaglavljivanju papira i redovnom čišćenju

Zatvaranja poklopca se ostvaruja prostim pritiskom na gornju površinu dok sa na začuja klik

Umetanje papira

Iako se papir u lasar može postavljati ručno list po list uobičajano ja da se ovo ipak obavlja automatski postavljanjem veće količina papira u kasetu Kasete uz lasar ja A4 formate ali se opciono mogu nabaviti i drugi formati koji su uobičajani na Zapadu U svakom slučaju u kasetu sa može postaviti isključivo papir koji po formatu odgovara deklaraciji kasala — nešto veći ili, češće manji formati papira zahtevaju



ručno postavljanja Za ovo su obazbađana dva vodica na poklopcu kasete čiji se razmak ručno podešava Vodica obezbeđuju da papir uvak ulazi centrirano u odnosu na osu štampača

Kasala se u štampač postavlja bukvalno „gurnjam" u predviđani otvor Nama jasnog znaka kada je postavljana u avoj koračni položaj ali sa razlika sa malo prakse lako uočava Najlakše ja kasetu postaviti lako što se njan zadnji kraj blago izdigna na gora — isda sa krajnji položaj pri umalanju najlakše uočava

Memorija, kertridž, interfejsi

Iapod kasete se nalaza dva velika otvora u čijoj sa unutrašnjosti otkrivaju konaktori U ovaj prostor se postavljaju lakozvani FONT CARTRIDGES — kasete sa pismima koje za sada po prilično valikoj ceni prodaja samo HP Sa dsana donja strana štampača nalazi se poklopac obezbeđan zavrtnjam koji krija konaktor za memorijsko proširanja Lasar se fabrički isporučuja sa 512 K memorija što je više nago dovoljno ako se štampač koristi kao kvalitatna pisača mašina Svaka ozbiljnija primana (ato na izdavaštvo pra svega) zahteva da se memorija proširi jednom od tri mogućnosti — sa 1 2 ili 4 M Cena avih dodataka nija mala — 1M se može naći za oko 600 do 1000 DM — ali bi ova dodatni trošak trabalo odmah uračunati u cenu pri odlučivanju o kupovini jar sa u mnogim situacijama bez dodatne memorija jednostavno ne može

Na zadnjoj strani porad uobičajane utičnica za mrežno napajanja i prakidača nalaza se i dva interfeja konaktora — centronika i RS232 Oba su ugrađena fabrički i mogu sa alternativno koristiti

Nakon uključivanja štampaču ja potrebna od nakolko sakundi do pola minuta da se zagraja i postane operativan (u zavisnosti od temperatura okolina) Najbolja ja odmah alartovati jednu od tri dijagnostička procedure Dirakina provera hardvera zaprijanosti uradaja i podešenosti sa postiže lasr procedurom koja se aktivira pritiskom olovkom sa lupim vrhom na laster TEST aakriven na donjoj lavoj strani štampača Nakon nakolko sakundi štampač traba da izbaci list sa lankim i ćlarim vertikalnim linijama

Testiranje uređaja

Sladeća dva lasla se aktiviraju sa kontrolnog panela i češće se koriste U oba slučaja prelhodno traba pritisnuti taster ON LINE kako bi štampač izgubio vazu sa računaram Prvi test sa dobija dužim pritiskom na taster obeležen sa PRINT FONTS/TEST (nakolko sakundi) Štampač će odštampati punu alfanicu znakova iz ASCII sala na način kako amo to već naviknli kod matričnih štampača ipak u gornjam delu će biti lapisano nakolko veoma važnih informacija — serijski brojevi operativnog sistema i ugrađanih fontova stanja svih opcija koja su izabrane u panal maniju i dve najvažnije količina memorija kojom laser raspolaže kao i broj stranica koje su odštampane od kako ja štampač napustio izbričku halu Brojač je nsaphodan za praćenje istrošenosti tonera Na ovom lsalu može sa uočiti još jedan važan datalj Čitava strana je ukovirana tankom linijom koja omeduja površinu na kojoj je lizički moguće štampanja Radna površina je za skoro pola centimatara manja sa svaka strana od atvarna veličine papira Poslednji tsal (koji to baš i nija u klasičnom smislu) je automatski ispis svih fontova koji su u tom trenutku na raspolaganju štampaču Pokreće se pritiskom na tall taster kao u prathodnom slučaju all ovaj put kratkim lapis sa dobija na dva stranice na prvoj su takozvani PORTRAIT (uspravni) fontovi dok su na drugoj odštampani kraći uzorci LANDSCAPE (pajzažnih oboranih) fontova Uz avaki font su date i njagove osnovne karakteristika

Povezivanje se računerom

Bez obzira koji od interfajsa izaberate za komunikaciju štampaču je praihodno potrebna postaviti odgovarajuće parametre komunikacije Ovo se ostvaruja prako kontrolnog panela

Kontrolni panal ima 7 naredbodavnih laztera i dva odvojana za oznakama +/— Najveći broj tastera ima dvojaku ulogu Funkcije lapisana u gornjam radu se dabija kratkim pritiskom na taster dok se funkcija iz donjeg reda poziva pritiskom i držanjam tastera nakolko sakundi Tasteri + i — služe za kružno kretanja kroz izbor vradnosti dok se izbor patvrđuje tasterom ENTER

Konkralno pri konfigurisanju cenovnih parametara rada treba izabrati MENU naradbu ali taster treba zadržati duže kao da se radi o funkciji iz donjeg reda (lako to na ovom tasteru nija posabno naznačano) Na displaju se pojavljuja SYM SET=ROMAN-8° što znači da je inicijalno izabran skup znakova koji u mnogome odgovara proširanom ASCII salu Pritiskom na + ili — tastera možemo pregledati i ostala varijanta ovog izbora i tastarom ENTER potvrditi naku od njih (ENTER oslavlja zvezdicu kao oznaku da je ta vradnost izabrana)

Sladeći pritisak na MENU taster daja izbor AUTO CONT=OFF pri čamu +/— nuda izbor ON ili OFF Radi se o načinu na koji će se štampač ponašati u slučaju kada naziare naka od mogućih grešaka u radu — stanja OFF zahteva da nakon uklanjanja grška pritisnama tazter CONTINUE ON stanja lara štampač će automatski nastavi rad

Još jadan pritisak na MENU taster nas dovodi do najvažnijeg izbora — tipa interfejsa kaji koristimo Tastarima + ili — bira sa paralelni ili serijski interfejs Ako se izabera ovaj drugi dalji prilaci na MENU daju mogućnosi da sa izaberu paramatri komunikacija Podržana su brzina od

300 600 1200 2400 4800 9600 i 19200 bauda.

Nakon što se prođe kroz sve opcije iz menija na displeju se pojavljuje 00 READY. Iz menija je moguće izaći i ranije pritiskom na ENTER taster.

Kratak pritisak na MENU taster otvara drugi meni kroz koji se kreće na isti način. Njegove opcije su vezane za radne parametre — izbor aktivnog fonta, broja redova na strani, način uvlačenja papira itd.

Sve izmene radnih parametara se trajno pamte i ostaju aktivne čak i kada se uređaj isključi. Međutim, u trenutku kada neki od parametara izmenite, nove vrednosti se ne primenjuju automatski. Nove vrednosti imaju efekta tek nakon pritiska na taster RESET i resetovanja štampača.

Upravljanje štampačem

Ako se štampač koristi kao pisaća mašina, za izbor pisma, formata papira, proreda itd. sasvim je dovoljan i kontrolni panel. Međutim, pravo korišćenje štampača zahteva poznavanje takozvanih ESC sekvenci kojima se laser može naterati da uradi bilo šta iz svog bogatog repertoara.

ESC sekvence koje je izabrao HP su danas jedini važeći standard stalističkog tipa naredbi za rad sa štampačem. Gotovo da ne postoji laserski štampač koji nije u stanju da simulira ovaj standard, čak i ako je operativni sistem ambiciozno zamišljen. Tipičan primer je štampač KIOCERA koji ima sopstveni, veoma bogat set naredbi, ali ga niko ne koristi! Najveći broj vlasnika automatski bira HP emulaciju.

Za upravljanje laserskim štampačem karakteristične su dve stvari. Prva je da se strana po kojoj ispisujemo u mnogome ponaša kao i ekran računara — moguće je u bilo kom trenutku i na bilo kom mestu odštampati bilo šta i tek nakon naredbe FORM FEED dobiti otisak na papiru. Višekolonski ispis, mešanje grafike i teksta i slične noćne more svih tekst procesora koji rade sa matričnim (linijskim) štampačima su pri ovakvom načinu formiranja strane dečja igra. Pomenuta osobina direktno proizilazi iz principa rada štampača koji mora imati u memoriji formiranu sliku strane pre nego što laserski zrak počne da je prenosi na selenski bubanj. Za drugu osobinu je direktno zaslužan HP — konačno i definitivno je izbačeno slanje parametara naredbi u binarnom obliku. Koliko puta ste se nervirali rastavljajući integer na sastavne bajtove da bi ga poslali štampaču? U HP naredbama svi parametri su u ASCII formatu i ESC (ASCII 27), FORM FEED (12) i LINE FEED (13 10) su jedini brojevi koji se i dalje šalju na stari način.

Na konkretnom primeru ćemo pokazati koliko je novi sistem jednostavniji. Pretpostavimo da treba na vrhu strane odštampati reč „RAČUNARI" a zatim i na dnu. Štampaču treba poslati sledeću sekvencu (ESC je označen sa Ec, ostali simboli se šalju kako su napisani):

Ec & a 1 R
Ec $ a 20 C
RAČUNARI
Ec & a 60 R
Ec $a 20 C
RAČUNARI

Prve i poslednje dve sekvence su zadužene za postavljanje kursora na željeni red i kolonu. Brojevi 1 20 i 60 se šalju kao ASCII znaci 1 2,0 itd. Ne znamo da li je to bila namera Pakardovih inžinjera ili je samo uspešni stekat, ali sistem je izuzetno pogodan za rad iz C-a. Pozicioniranja kursora se veoma zgodno izvodi printf naredbom

```
fprintf(prt, "/033&a%dR/033&a%dC", x,y)
```

gde su X i Y celobrojne promenljive. Situacija nije ništa komplikovanija i u ostalim programskim jezicima.

Sve naredbe su podeljene u 6 grupa: formatiranja strane, pozicioniranja kursora, fontovi, grafika, makroi i ostale naredbe. Naredbe su tako izabrane da se štampač može tretirati na dva sasvim različita načina. Prvi imitira linijske štampače — postavljaju se margine, linijski razmak i kursor pozicionira u jedinici veličine slova — pri normalnom proredu na papir staje uobičajenih 66 redova sa po 80 ili 132 znaka.

Ništa ga ne može iznenaditi

Drugi način je daleko interesantniji. Ako je prvi ekvivalentan tekst režimu video adaptera, onda je ovaj drugi pravi grafički režim — kursor se pozicionira na bilo koju tačku (rezolucija 1/300 inča), postoje naredbe za popunjavanje pravougaona površine različitim gustinama i tipovima rastera (istim naredbama se povlače horizontalne i vertikalne linije) i naredbe za rad sa grafikom različite gustine — 75, 100, 150 ili 300 tačaka po inču. Pri slanju bit mapiranih slika čeka vas još jedno prijatno iznenađenje — slika se šalje red po red (kao i na ekran visoke rezolucije), pa više nema potrebe za grupisanjem osam redova i preračunavanjem vertikalnih grupa bitova kao kod matričnih štampača.

Setu naredbi se zaista ne može ništa prigovoriti. On je optimalno izabran i štampač se gotovo idealno ponaša i u nepredviđenim situacijama. Ako postavite kursor van površine papira tekst koji štampate se jednostavno ignoriše — ako izabere pismo nepostojećih karakteristika, štampač aktivira ono koje najviše odgovara zadatoj specifikaciji. Sve u svemu, prava je šteta što se laserski štampači (ili bar način upravljanja) nisu pojavili ranije — kada matrični štampači budu potisnuti u istoriju, tekst procesori će sasvim drugačije izgledati! Najveći broj korisnika nabavlja laserski štampač da bi ga upotrebio za kvalitetnu pripremu štampe uz pomoć nekog od programa za prelom, ne razmišljajući da bi i sami mogli da preuzmu upravljanje. Ipak, čak i u ovom slučaju savetujemo da se malo zavirí u set naredbi HP lasera — ustanovićete da se mnogi teški prelomi mogu veoma lako izvesti kratkim programima za konkretnu namenu.

Oblici slova

Rad sa pismima je zaista priča za sebe. Fabrički su ugrađena praktično samo dva pisma — *Courier* i *Line Printer* su neproporcionalna pisma koja u mnogome podsećaju na ona sa klasičnih pisaćih mašina. Ovaj drugi tip je sasvim upotrebljiv — slova su manja (oko 140 slova staje u jedan red), deluju veoma čitko i često se koristi za pripremu listinga i sličnih tekstova za štampu (veliki broj zapadnih kompjuterskih časopisa i knjiga koristi upravo ovaj font za listinge programa). Za *Courier* nam nije jasno zašto je izabran kao osnovno pismo koje se fabrički ugrađuje — slova su prilično krupna, potreban je veći prored da bi tekst bio čitak, pa ne možemo da zamislimo ni jednu pažnje vrednu primenu ovog pisma. Svakako da se radi o čisto ekonomskom rezonovanju — pravi fontovi se dodatno kupuju, a kada budete videli bilo koji od njih brzo ćete zaboraviti na ono što je ugrađeno u štampač. HP ipak zaslužuje kritiku za ovakav potez. Sva popularnija klasa POSTSCRIPT štampača ima, na primer, ugrađen kao minimum dva osnovna pisma koja se najčešće koriste — helvetiku i tajms. ROM od nekih pola ili ceo megabajt sa ova dva pisma ne bi morao drastično da poveća cenu HP uređaja.

Bez obzira na ugrađene fontove, rad se uglavnom odvija sa takozvanim „SOFT" fontovima — datotekama koje sadrže definiciju pisma za svaku željenu veličinu posebno. Izbor iz HP kataloga je za sada relativno skroman (Helvetica, Times, Garmond i nekoliko naslovnih pisama većih gradacija), ali dobro izabran. Cena jedne diskete na koju staje jedno pismo u 7 ili 8 gradacija se kreće od 400 do 800 DM.

Osnovna karakteristika soft fontova je da definišu svaki znak kao bit mapu, što ima nekoliko posledica — svaka veličina ili varijacija (bold, italik) zahteva poseban font (datoteku) i fontovi većih gradacija mogu zauzeti prilično memorije štampača (Times 24 punkta je zapisan u datoteci od oko 110K). Samo zauzeće memorije nije toliki problem kao što to može biti vreme potrebno da se font pošalje štampaču.

Kertridži sa ROM fontovima su veoma privlačno rešenje koje, na žalost, ima i jednu neprijatnu manjkavost — nije nam poznato kako bi se izvelo ubacivanje naših slova, koje se inače kod SOFT fontova lako izvodi nekim od font editora. S obzirom da se kod nas *Xerox Ventura Publisher* program najčešće koristi za prelom, moramo da upozorimo potencijalne kupce kertridža da sadašnja verzija programa nisu u stanju da rade sa fontovima iz njih.

O dobrima sve najbolje

Pakardovom štampaču se zaista može malo šta prigovoriti. U ponašanju operativnog sistema nismo primetili ni jednu grešku.

Postoji međutim nekoliko detalja oko mehanizma koji uvlači papir koji mogu da prave probleme u radu. Prvo da kažemo da smo isprobali štampanje na različitim podlogama — kvalitetan bankpost papir je dao najčistiji otisak, a odmah za njim i papir namenjen foto-kopir mašinama. Laser ostavlja dovoljno zapečen otisak na papirnim matricama, što ga čini posebno pogodnim za ovu vrstu priprema malotiražnih izdanja. Paus se može dobro koristiti pod uslovom da nema masnih otisaka na kojima se toner slabije hvata. Najgora situacija je, na žalost, sa foto-kopir folijama koji bi mogle da zamene grafički film — isprobali smo nekoliko vrsta dostupnih na našem tržištu, ali se otisak na većim zacrnjenim površinama povlači ka centru ili ivicama na sličan način kao kad se tuš prospe na površinu koja ga slabo upija. HP folije namenjene baš ovom štampaču nismo imali prilike da proverimo.

Folije i papirne matrice se najčešće, zbog neodgovarajućeg formata, ulažu u laser ručno. Iako se ubacuju preko vodica na kaseti za papir, prihvata ih isti mehanizam koji papir izvlači iz same kasete. Da bi radio, štampač mora imati bar jedan papir u kaseti, koji se onda polako povlači zajedno sa listom koji se ulaže ručno pa dolazi do zaglavljivanja.

Održavanje štampača je izuzetno lako. Toner kaseta je deklarisana na 4000 kopija i u praksi se pokazalo da je ova vrednost vrlo precizno izmerena — gotovo nikad nije manja, dok je ponekad za samo nekoliko desetina kopija veća. Na stotinjak kopija pre kraja štampač na displeju povremeno ispisuje upozorenje „TONER LOW".

Pri ubacivanju svežeg tonera treba pregledati unutrašnjost štampača. Eventualne nečistoće transportnog mehanizma treba ukloniti mekom krpom, dok se sa leve strane nalazi mala četkica kojom se čisti zaprljana elektrostatička nit (CORONA WIRE). Sa čišćenjem ne treba ni malo žuriti — i nakon nekoliko potrošenih kaseta sa tonerom bez čišćenja neće biti nikakve razlike u kvalitetu otiska.

Na kraju da kažemo i nešto o uputstvima koja su priložena uz štampač. Prvo je nevelika knjižica naslovljena sa „Getting Started with LaserJet series II" (kako početi rad sa štampačem) i pruža samo neophodne informacije o instalaciji, konfigurisanju i priključivanju na različite računare. Druga knjiga, „User's Manual" (korisničko uputstvo) pruža opet na elementarnom nivou objašnjenja osnovnih pojmova, nekoliko važnijih naredbi (Esc sekvenci) i način prilagođavanja različitih programa za rad sa ovim štampačem (WordPerfect v4.1, DBase III, MS WORD v2.0 i v3.0, WordStar itd). Uputstvo je zaista na elementarnom nivou, iako je data veoma pregledna i kompletna lista svih naredbi koje štampač raspoznaje, nedovoljno je da se iskoriste čak i neke jednostavnije opcije. Detaljno obrazloženje (a ono je stvarno potrebno, pogotovo kada je rad sa fontovima u pitanju) ostavljeno je za „Technical Reference", koji nismo imali prilike da vidimo. Snalazili smo se koristeći priručnik za raniji model HP LaserJet Plus koji u potpunosti odgovara, iako jasno nedostaje više naredbi kojih na ovom modelu nema.

Laserski štampači su gotovo od prve preboleli dečje bolesti novih uređaja koji se pojavljuju na tržištu. S obzirom da veliki broj proizvođača koristi mehaniku veoma malog broja proizvođača, kvalitet otiska se malo razlikuje od uređaja do uređaja. Cena HP LaserJet Series II štampača varira i razlikuje se od jedne konfiguracije do druge. Pakarda u Jugoslaviji zastupa Hermes, 61000 Ljubljana, Celovška 73, 061-552-941. Predstavništvo u Beogradu, Zrmanjska br. 10, tel. 011/557-282, 557-234.

Zoran Životić

Dragan Kopunović, vlasnik softverske firme C.H.ANGE

Znanje se uvek isplati

U pionirskim danima razvoja domaćeg računarstva san svakog mladog programera bio je da napiše dobar program i da ga dobro proda u inostranstvu. Iako je nekoliko najvrednijih i (možda) najarečnijih hakera uspelo da ostvari svoje snove, vreme je raspršilo mnoge romantične ambicije — pokazalo se da je dobar program teža prodati nego napisati. Zato verujemo da će mnogi programeri biti zainteresovani za saradnju sa kanadskom softverskom firmom C.H.ANGE i njenim vlasnikom Draganom Kopunovićem, koji je tokom maja i juna, kao gost Privredne komore Jugoslavije, boravio u našoj zemlji u potrazi za novim programima i pregovarao sa vodećim radnim organizacijama koja se bave proizvodnjom i plasmanom softvera. Jedan od sagovornika bila je i redakcija „Računara".

Da počnemo sa malo istorije – pošto je diplomirao matematiku na beogradskom Prirodno-matematičkom fakultetu *Dragan Kopunović* se otisnuo u dalaki Toronto (Kenada) u kome je pre četiri godina osnovao firmu sa sugestivnim nazivom C H ANGE. Prve delatnosti firma C H.ANGE bile su konsultacije, planiranje metodologije i planiranje informacionih resursa — uspostavljena je dakle saradnja sa velikim korisnicima koji već imaju komplikovane računarske sisteme i brojne aplikacije a zelim su ti programi i podaci povezani u međusobne veze koje će razultirale novim kvalitetom primene računara. Glavni klijenti firme C H.ANGE bile su dve kenadska ministarstva i dve velike naftne kompanije.

Početkom 1985 godine C H.ANGE započinje prodor na tržište PC softvera uameravajući svoju delatnost na dizajn razvoj dokumentacije marketing i podršku tj. na sve osim samog razvoja saftvere koji je na Zapadu izuzetno skup. Poznajući kreativne potencijale naših zemljaka *Dragan Kopunović* je nedavno kao gost Privredne komora posetio Jugoslaviju u potrazi za originalnim PC programima koji bi mogli da se plasiraju na Zapadno tržišta.

Nema nepoželjnih tema

Uzevši u obzir da na tržištu postoji ogroman broj PC programa, kolika su uopšte šanse za uspešan plasman jednog programa među desetinama sličnih? Da li je tržište zasićeno?

— Ne tržištu sa zaista nalezi mnogo PC programe eli ih je malo kvalitetnih. Predsednik multinacionalna kompanije *General Motors* je nedavno izjavio da su po njegovom videnju stvari dve glavne prepreke razvoju svetske ekonomije — nestabilno i teško predvidljivo tržište nafte i loš softver kojim je teško ovladati i u koje se ne može dovoljno pouzdati. Tržište je dakle isuviše otvoreno za dobra programe!

Da li se to odnosi i na softverska paketa standardna, namena kao što su na primer tekst editori?

— Mi se na bojimo konkurencije što znači da bismo se eko verujemo da je program koji rešave neki standardni problem u nekim aspektima bitno bolji od postojećih upustili u njegov marketing.

Kako treba da izgleda jedan softverski paket da bi uspeo na tržištu?

— U centru je naravno dobar program eli se tu priča ne završave. C H.ANGE primenjuje standardnu metodologiju za dokumentovanje softvera koje podrazumeva leporuku tri priručnika uz svaki paket. Prvi priručnik je zapravo generalni opis problema koji se rešave prilegođen ljudima koji nemaju posebna računarske iskustva eli imaju potrebe za rešavanjem tog problema. Drugi priručnik je uputstvo za upotrebu programa tj. tekst o tome kako se problem opisan u okviru prvog uputstva rešava primenom računara. Treći priručnik je nejzed referentne prirode i detaljno opisuje sveku naredbu koju program podržave.



U potrazi za novim programima: Dragan Kopunović

Koliki tim ljudi priprema ovu dokumentaciju i koliko ta priprema traje?

— C H.ANGE ima tri stalno zaposlena računaraka stručnjeka četiri pisca tehničke dokumentacije *(technical writer)* i dve dizajnere po potrabi se angažuje još autora III stručnjaka. Uzevši u obzir i vreme potrebno za oglašavanje između finalizacije samog programe i njegovog lektičkog izlaske na tržište prode četiri do šest meseci zavisno od prirode samog programa.

Programer kao svetinja

Kako zamišljate saradnju sa jugoslovenskim autorima programa?

— Autor zadrževajući vlasništvo nad programom uetupa firmi C H.ANGE ekskluzivno pravo da kompletira raklemira i prodaje programski paket u čitavom svatu osim Jugoslavije. Određeni procenat od cene svakog prodatog primerka pripede autoru.

Koliki je to procenat?

— Procenat zaviei od posle koji je firma C H.ANGE uložile u kompletiranje proizvode — eko se radi o igri za koju je trebalo napisati samo kratko uputstvo i pripremiti pakovanje procenat je veći eko se radi o paketu za koji je trabalo pripremiti opsežnu dokumentaciju naši troškovi su bili mnogo veći eli je i cena programa viša. Konkratni detalji dekle zavise od samog programe.

Kako se rešava problem autorskih prava?

— Kada posle prvih kontakata sa autorom zaključimo da je ponuđen program vredan enelize potpisujemo ugovor o čuvenju poslovne tajne koji za autore predstavlje evojevrsno obezbeđenje — ukoliko se iz bilo kog razloga delj aranžmani ne ekispa firma C H.ANGE je dužna ne samo de uništi primljena primarke programe već i da u meri u kojoj je to razumno ne eksploatiše ideju ne kojoj je ponuđeni program zasnovan. Docnija ugovori koje ćemo eklepati poeredstvom novosedske advokatske kancelarije koje nee zestupa obezbeđuju kako zaštitu autore leko i našu zaštitu. Treba reći da je medu softverskim kompanijeme ehvetenje autorskih prave i uopšte poslovnog morale uglavnom veome ozbiljno pošto mi živimo od autora programe — firma koje na bi etriktno poštovale preve autora bi u najkraćem roku propalal. Ponekad se međutim javljaju odrađeni problemi medu samim autorima.

Šta autori programa treba da učine da se takvi problemi ne bi pojavljivali?

— Problemi mogu da se jave eko na nekom programskom paketu redi više ljudi pri čemu njihove međusobne preve nisu uređene. Tada se može dogoditi da jeden od autora prode naku verziju paketa jednoj e drugi drugoj kompaniji što izazive veome najsana sudske sporove koji sa otegnu toliko de programi zastarl pre izlaske ne tržište. Zato je neophodno da se pre početka

timskog rada na nekom projektu precizno i po mogućnosti napismano uređe prava svakog pojedinog saradnika i da se takvi ugovori ako razvoj duže traju obnavljaju posle svakih nekoliko meseci

Softver na probu

Koliku pratnju za plasman programskih paketa predstavlja softversko piratstvo?

— Softversko piratstvo je naravno, prisutno Pokazuje sa medutim, da su vrlo retki ozbiljni korisnici softvera koji bi sa odlučili da programski paket koji im je zaista potreban piratuju — na taj način na ostaju samo bez originalne dokumentacije već i bez itekako značajne podrške proizvođača softvera Ispravke nova varzija telefoneka i plemena pomoć pri instalaciji i akaplosteciji

A piratstvo koje se svodi na pregled brojnih programskih paketa da bi se izabrao onaj koji je stvarno potreban?

— Među pojedinim korisnicima loga svakako ima lola ozbiljnija kompanija medutim kupuju po jedan primarak svakog makar na prvi pogled interesantnog paketa kako bi njihovi stručnjaci ustanovili da li se radi o programu koji bi sa mogao korisno upotrebiti Nija čak laključeno da se nakoliko hiljeda primaraka korektno napisanog i stručno reklamiranog programa (što je sasvim dovoljno da se čitav projekat pokrilja) proda ovakvim potencijalnim korisnicima od kojih mnogi uopšla neće odlučiti da program docnije stvarno koriste

Kakvi programski paketi trenutno imaju najveće tržišne izgleda?

— Firma C H.ANGE ja najviše zainteresovane za programa koji na neki način zamenjuju na Zapadu veoma skup tjudski rad ekspertni slatami prilagodani nakim oblastima tahnike generatori aplikacija- usluzni program, konvarzija izmedu različitih formate komunikcija mikro računara sa većim sistemima Zaintaresovani amo neravno i za OS/2 aplikacija pošto će ove trenutno relativno slobodno tržišta u toku sledaćih godina postati itekako značajno

Rizik bez rizika

U koliki se po vašem mišljenju rizik upušta potencijalni autor programa kada odluči da uloži značajan trud u razvoj neka aplikacije?

— Najznačajniji parametri uspeha nekog programa su originalnost ideja i kvalitet realizacija ali naravno postoja i mnoga druga okolnosti koja su svakako učinile da naki objektivno kvalitetni programi na uspeju na tržištu Varujem, medutim da se znanja stečano tokom razvoja programa mora isplatiti — ukoliko nako u današnjam trenutku počna da priprama neku OS/2 aplikaciju i tom prilikom dobro upozna novi operativni sistem, sigurno je da će mu sa to znanja isplatiti kroz buduće poslova čak i ako sama aplikacija na uspe na tržištu!

I za kraj kakve ste utiske stekli tokom upravo okončana svojevrsne turneje po Jugoslaviji?

— Posatio sam više valikih radnih organizacija u Novom Sadu Seogradu Zagrabu i drugim msatima i vodio interesantne razgovora o saradnji Nisam medutim imao mnogo prilika da kontaktiram sa nezavisnim autorima softvera koji, kada sa radi o paketima srednja valičina mogu najviše da ponude Varujem da je ovaj razgovor sa Računarima prilika da se i ts manjkavost ispravi

Rasgovor ja svakako prilika za tirmu C H.ANGE da se abrati nezavisnim autorima softvera ali ja i izvanredna prilika za te autore da svoja dostignuća komercijalno plasiraju Ukoliko ste, dakle sastavili naki PC program koji ja po vašem mišljenju interesantan za širi krug korisnika III imate naku ideju koju biste u bliskoj budućnosti transformisali u program obratite sa firmi C H ANGE A onda ko zna Vredi pokušatil

I za kraj prava stvar C H ANGE 7 Jackso Avenue Suita 407 Toronto Ontario Canada

Dejan Ristanović

Redakcija „Računara", Institut „Mihajlo Pupin" i „Magmedia"
Samogradnja „tima 011"
Prijava samo do 1. septembra

Počela isporuka štampanih kola

Čitaoci koji su nam na prvi poziv dostavili preliminarne narudžbenice sigurno su bili veoma iznenadani kada su sredinom juna, samo petnaestak dana nakon odmrzavanja cene i devalvacije dobili pismanu ponudu za sanovni komplet za samogradnju „tima 011" U znak zahvalnosti što su svojim glasovima ohrabrili redakciju da organizuje akciju samogradnje potrudili sme sa da uz prioritet u isporuci obezbedimo i povoljne cene Zahvaljujući razumavanju naših saradnika u ovoj akciji Magmedijl koje je obezbedila diskete za sistemski softver pod izuzetno pavoljnim uslovima i preduzeću Kompjutranika iz Skopja koje je u znak podrške akciji proizvelo inicijalnu količinu štampanih kola gotovo bez ikakve zarada ova cena je iznosila samo 49000 dinara Nisu loše prošli ni čitaoci julskog broja Računara — njima je pružana prilika da do 20 juia nabave komplel po ceni od 52000 dinara

Uvažavajući desta nesrećne okolnosti u kojima stertuje ova akcija — period lainjeg raspuste i godišnji odmora odlučili amo da produžimo rok uplate do 1 septembra Cena au na žaiicel znatno nepovoljnija all realnija lako je odziv više nego ohrabrujući imamo utisak da sa čitaoci i ovoga puta prave dvouma U svim projektima koje smo imali de sada izuzev kod računara galaksija Interesovanja za akciju je počinjalo da rasta tak kada i redakcija zakljući Cena kapitala je danas izuzetno velika a držanje zaliha izuzetno skupo i redakcija nema ekonomskog interesa da investira u valiku proizvodnju štampanih kola Ta činjanica nas nevodi da čitaocima ograničimo vrama za razmišljanje i rokove u kojima se može naručiti ganovni komplet sa garantovanom isporukom U ovam času procenjujamo da će ovaj deo akcija bili zaključen 1 septembra Nakon toga ćemo čitaocima staviti na raspolaganja samo onoliko kompleta koliko preostana ako uopšte preostane iz već naručenih količina

Iako u ovom času vać više od 400 samogradilelja tima 011 gori od nestrpljenja da popuni svoje pločice mi još uvak nemamo ponudu za kompiel delova iz inostranstva kaju bismo mirna savesti mogli da preporučimo čitaocima Cene mamorijalnih čipova još uvek divljaju probijajući sve praipostavke i kalkulacije a kojima amo ulazili u ovu akciju

Plašeći sa da nam ipak na lipše magara de zalane trava planirama da definitivnu odluku donesemo do sredina avgusta i da je objavimo u septembarskom broju Teši nas jedino činjanica da medu samograditeljima tima 011 ima okoralih hardveraše koji znaju kako sa dolazi do čipova i bez pomoći redakcije

Računar tim 011 sa u varzlji za samo gradnju sastoji od tri štampana kola dvoslojna matična ploče ploče za ispravljač i interfejsna pločica za PC tastaturu Štampana kola su uradene prema profasionalnim standardima — na vitroplastu presvučena lakom a sa gornja strane je štampan raspored komponenata

Narudžbenicom u ovom broju mogu sa naručiti sledeće komponante

- Dvoslojne matične ploče za tim 011 i magmedia diskata sa sistemskim softverom i odgovarajućim drajvarima po ceni od 58500 dinara
- Jednoslojno štampano koto za ispravljač tim 011 sa štampanim rasporedom komponenata po ceni od 18000 dinara
- Jednoslojne štampano kolo za interfejs za PC tastaturu sa štampanim rasporedom komponenata po ceni od 6000 dinara
- Komplat štampanih kola za tim 011 — matične ploče ploče za ispravljač i pločice interfejsa za PC kompetibilnu tastaturu — i magmedia diskata sa sistemakim softverom i odgovarajućim drajvarima po ceni od 73500 dinara
- Torusni transformator tovarnik 80 VA sa priborom za montažu za ispravljač po povišatenoj ceni od 25000 dinara Transtormator je praizvedan specijalno za projekat koji objavljujamo u ovom broju i dvostruko je jatinlji od sličnih medala u Nemačkoj ili Englsakoj Zahvaljujući ovako pošoljnoj ceni samogradilelji tima mogu da dodu do jeftinog ispravljača na samo za svoj računar nego i za ostala projekta Isporuka pouzećem rok 15 dana od prijama narudžbenice

Cena za kombinaciju štampano kola/sistemska diaketa važe samo za ona koji uplatu izvrše unaprad na žiro-račun RO SIGZ 60802--503-23264 sa naznakom Računari — štampano kolo i to najkasnija do 1 septembra Uz narudžbenicu je obevezno dostaviti i fotokopiju uplatnice Isporuka dve do tri nedalje po prijamu uplatnice Torusni transformator i praktički ispravljač naručuju sa pouzećem

Narudžbenica

Obaveštavam vas da sam dana ____ izvršio uplatio na žira-račun RO SIGZ 60802-603-23264 sa naznakom Računari — štampano kolo iznos od ____ za sledeće komponanta

1) Matična ploča — sistemske diskete po ceni od ____ 58500
2) Ploča za ispravljač ____ 18000
3) Interfejsna ploča za tastaturu ____ 6000
4) Komplet štampanih kola — sistemska diskata ____ 73500

Dokaz o uplati dostavljam vam uz ovu narudžbenicu

Molim vas takoda da mi pouzećem pošiljate sledeće elemente

1) Torusni transformator tovornik 80 VA ____ 25000

Poštanski troškovi padaju na moj larat a potrabnu sumu ću ispisiti poštaru prilikom prauzimanja pošiljke

Ima i prazima ____ Poštanski broj i
mesto ____ Ulica i
broj ____

Datum ____ Svojaručni potpis

Narudžbenicu sa uplatnicom dostaviti na adresu Računari — za „tim 011 11000 Beograd Bulevar vojvode Mišića 17

WordPerfect 5.0

Niko nije WordPerfectan

WordPerfect već godinama zauzima oko 30% tržišta i sigurno je najpopularniji program za obradu teksta. Odavno je poeteo dobar, odličan, pe i sveobuhvatan procesor reči (verzije 4.2), ali — zahtevi se menjaju. Do sada je većina programa za pisanje verno podražavala mogućnosti pisaće mašine, jedino je izvedba bila elektronska i, kao takva, nosila se sobom niz prednosti. U poslednje dva godine apajanja reči i grafike na ekranu računara postalo je prvo mogućno, a zatim i naminovno. Osim specijalnih programa za stono izdavaštvo, pojavili su se i procesori reči koji mogu da „ukradu" sliku sa ekrana i umetnu je u tekst (TotalWord, GEM Write, Windows Write, Wordstar 2000 ver 3.0), a nešto slično može i MS Word uz pomoć dodatnih programa. PageView. Verzija 5.0 je, dakle, odgovor WordPerfect-a na zahteve vremena.

WordPerfect je previjan za iskusne korisnike — prevodioca sekretarica advokate i, uopšte, sve profesije koje su vezane za puno pisanje. Takvim korisnicima verovatno neće smetati neka nelogičnost — pod uslovom da se česte operacije izvode brzo. Zato su svi funkcijski tasteri opterećeni maksimalno i to na veoma čudan način. Preciznija naredbe su proizvoljno vezane za tastere. Svaki sivi taster sadrži po četiri funkcije (sam i zajedno sa tasterima **Alt**, **Ctrl** i **Shift**) ali u nekim slučajevima na vidi se nikakva veza niti funkcije jedne sa drugom niti po parovima tastera. Tako je na F7 vezano sledeće: **Exit, Math/Columns, Print** i **Footnote**, a tasteru F5 pridružani su **List Files, Mark Text, Date/Outline** i **Text In/Out**. Korisnik *WordPerfect-a* mora da ima odlično pamćenje a to se jedino postiže svakodnevnim višečasovnim radom (pisanjem prekucavanjem tekstova i slično)

Za ljubitelje originala . . .

Kretnja kursora po tekstu je originalno. Kursori se za po jedno slovo pomeraju uobičajenim kursorskim strelicama (brojevi 4 6 2 i 8 na numeričkom delu tastature) a kombinacije tastera Ctrl i leve i desne strelice premašta kursor za po reč ulevo odnosno, udesno. Tasteru **Home** pridaje se veliki značaj. Prilisak na **Home** je ulaz u (nevidljivi) meni sledeći prilisak na neke tastere ima posebno značenje. Tako **Home-Backspace** briše reč ulevo. **Home- strelica-u-vis** lista teksi sa po jedan ekran unetrag (nagore). **Home- „strelica-u-levo"** liste tekst za po jedan ekran unagrad (nadole), irostruki niz **Home-Home- „strelica-u-vis"** vodi kursor na početak datoteke i tako dalje. Sivi **plus** i **minus** takođe listaju tekst za po jedan ekran nadole i nagore. Tasten **PgUp** i **PgDn** vode na početak prathodne i sledeće strane. Taj izbor je pomelo nepriroden jer korisnici obično razmišljaju u terminima onoga što vida — a veličina ekrana je skoro uvek znanja od lizička veličina stranice. Kombinacija **Ctrl-Home** je naredba direktnog preiska na neki deo teksta (Goto). I na ovaj način može se doći na prethodnu i sledeću strenu, prethodni i sledeći stubac prvi i poslednji stubac na početak strane sa datim rednim brojem, na sledeće pojavljivanja nekog slova u okviru sledećih 2000 znakova na prehodni položaj kursora, na početak bloka ako je blok osvetljen i tako dalje. Na taj način *WordPerfect* omogućava sve korisne iako možda nestandardne pokrete kursora po tekstu.

Tester **Esc** označava broj ponavljanja sledeće radnje. Podrazumeva se vrednost 8 teko da niz **Esc** i **„strelica-na-dole"** pomere kursor za osam redova nadole

Interesantno je napomenuti da su kombinacija **Ctrl-End** i **Ctrl-PgDn** destruktivne. **Ctrl-End** briše sva slove do kraja reda a posle **Esc** briše naznačeni broj redove ispod položaja kursora. **Ctrl-PgDn** briše tekst od položaja kursora do kraja strane. Za svaki slučaj korisnik mora da potvrdi tekvu odluku

Poslednja značajna kombinacija, **Ctrl-PgUp**, vezana je za makro varijabla a ne za pokrete kursora ili izmena teksta

Korisnicima koji poznaju druge procesore reči ovakav raspored tastera biće neobičan pa i opasan. Na primer, skoro svi ASCII editori i procesori tekste za tastere **PgUp** i **PgDn** vezuju pomeranje kursora za veličinu ekrana a **Ctrl-PgUp** i **Ctrl-PgDn** obično voda na početak i kraj teksta. Zato je najbolje na mešati procesore reči tj odmah se odlučiti za jedan moćan program i samo njega koristiti

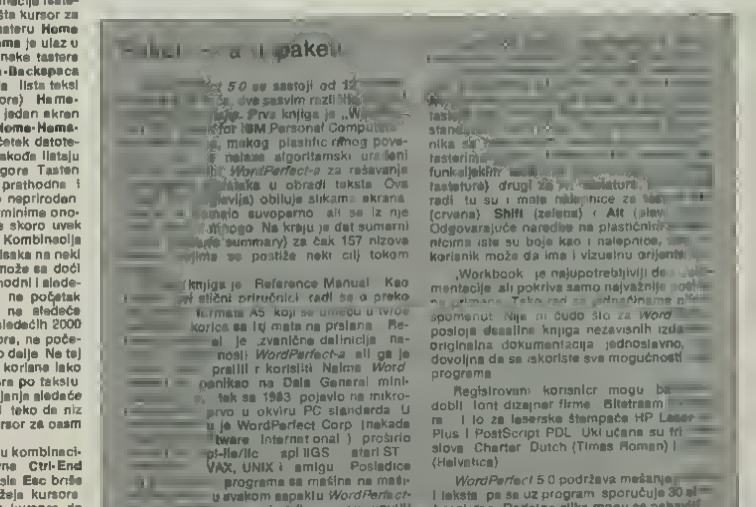
i ljubitelje klasika

Korisnik izdaja naredbe kroz sistem menije. Meni se pojavljuje pri dnu ekrana a svaka opcije je numerisana rednim brojem od 1 pa naviše. Uvak treba otkucati i redni broj opcije i pritisnuti Enter. Nula uvek izbacuje iz manija i najčešće sa upravo ta vrednost podrazumeva ako se odmah pritisne **Enter**. Osim po rednim brojevima svaka opcije se može birati i nekim slovom. To slovo se ističe pojačanim intenzitetom ali je izbor slova često nelogičan

WordPerfect ne radi sa danas modernim i opšteprihvaćenim padajućim manijima prozorima ili mišem. Takve ideje došle su prakasno za ovaj program. Sa druga strane, meniji po rednim brojevima jedino su logičan izbor ako se želi prenosivost programa a taj način komunikacije osobito je pogodan za makro naredbe

Premeštanje kopiranje i brisanje teksta vrše se iz menija **Ctrl-F4**. Korisnik može da odabere količinu teksta na dva načina. Kombinacija **Alt-F4** označava početak bloka (istovremeno sa na dnu ekrana naizmence pali i gasi poruka **Block On**. Pomeranja kursora osvetljava tekst i definiše sam blok. **Ctrl-F4** tada omogućava klasične operacije **Copy, Move, Delete** i **Retrieve** (još jedno korišćenje označanog teksta). Korisnik more da dovede kursor na mesto gde blok treba da se pojavi i da onda pritisne Enter. Ako blok nije prethodno definisan **Ctrl-F4** pruža mogućnost da se osvetli čitava rečanica pasus ili strane a dalje je postupak sličan kao da je korisnik sam odabrao blok teksta

Jedna izvanredna osobina *WordPerfect-a* je povratak obrisanog teksta. To svojstvo sada





lipa WPK. Ona sadrže akcije pridodeljene tasterima pa je moguće kompletno predefinisati tastaturu. (Ovu ideju *WordPerfect* je preuzeo od programa *XyWrite III Plus*.) Uz sem program isporučuju se tri WPK datoteke i najduža od njih MACROS.WPK odlična je poleđina lačka za stvaranje ličnih tastature.

Makro naredbe

Makro naredbe i skraćenice oduvek su bile jaka strana *WordPerfect-a*. Prilikom na Ctrl-F10 počinja animanja pri čemu se snima makro naredbe ako se prilena na Alt istovremeno sa nakim tasterom a ako se upiše jedno ili više slova — snima se skraćenica. Samo Enter snima privremeni makro (biva uništen po izlasku iz *WordPerfect-a*). Kasnije se pritiskom na Alt i neko slovo animijeni tekst odmah ubacuje a za aktiviranja skraćenica treba pritisnuti Alt-F10, navesti skraćeno ime i završiti ga se Enter. Jednom animijena makro naredba može se menjati posebnim makro editorom. Svakoj makro naredbi odgovara tačno jedna datoteka na disku se prazinanom WPM.

Više makro naredbi može se izvršiti kombinovanjem se Esc, ili komponovanjem niza makro naredbi. Moguće je čak koristiti i makro naredbu u makro naredbi.

Sva ove mogućnosti poseduju i drugi programi, kako posebni makro generatori tako i procesori reči. Međutim makro naredbe su u *WordPerfect*-u razvijena do pravog programskog jezika. Pre svega postoji deset varijabli označenih ciframa od 1 do 0 (gledano s leva na desno na tastaturi). Kombinacija Ctrl-PgUp dozvoljava da se u jednu od tih deset varijabli smesti proizvoljan tekst (običan ili program) a kasnije se se Alt i cifarskim tasterom taj sadržaj varijable unosi u tekst odnosno, izvršava se ako je bio program u pitanju. Ako je prethodno bio označen blok na ekranu posle Ctrl-PgUp moguća je sadržaj bloka pridodaliti nekoj kombinaciji tastera Alt i jedne od deset cifara.

Nad varijablama se mogu vršiti poređenja pridodeljivanja i logička operacija AND i OR.

Makro jezik sadrži sve elemente programskog jezika: komentara, brz izlazak iz IF naredba, poziv potprograma u okviru same makro naredbe, naredbu CASE, slanja poruka korisniku, uključivanja i isključivanja promena na ekranu dok se izvršava makro naredba ELSE i ENDIF kao delove IF naredbe obezbeđa naredbi i GOTO izvršavanja neke druge makro naredbe (do 20 poziva po dubini) izvršavanja korak po korak i slično. Ovako moćan makro jezik pratvara i sam program *WordPerfect* u specijalizovani interpretativni jezik visokog nivoa. To nipošto ne znači da se mora poznavati programiranja da bi se pisalo u *WordPerfect*-ul. Za svakodnevno pisanja obična skraćenice su više nego dovoljna jer svaki tekst se lonsko sastoji od nekoliko desetina česlo spominjanih termine. Međutim makro naredbe kao programski jezik pružaju dodatne mogućnosti. Poslojl čak jedna firma koja prodaje slotinak najvažnijih makro naredbi za *WordPerfect* i to je po svim elementima idenlično predaji softvera kao u bejzlku ili peekalu.

Makro naredbema se jedino može zamariti da se izvršavaju sa diska a ne iz memorija. Za ulahu ni konkurencija nema nlšta bolja, osim XyWrite-a — koji do 38 makro naredbi izvršava direktno iz memorija.

Prozori i interni kodovi

WordPerfect može da radi sa najviše dva horizontalna prozora. To je minimum jer *Wordstar 2000* može da radi se tri *MS WORD* se osam a *XyWrite* i *Note Bena* sa čak devet horizontalnih i vertikalnih prozora istovremeno na ekranu. No i ta dva prozora su dovoljna za osnovna poslupka popul kopiranja premeštanja ill prosiog reeglednja dva teksta odjednom. Mnogo zanimljiviji prozor aktivira se naredbom Reveal Codes (Alt-F3). Naime svi procesori reči ubacuju u tekst interna kodova sa različite vrste slova, caninra- ▶

Imaju praktično svi procesori reči— ali ovde program pamti više brisanja izvedanih na lati način kao jedan lekst pa ako obrišete pet reči uzeslepca npr. kombinacijom Ctrl-Backspace mači ćete da u tekst vralite svih pet reči kao jednu celinu. Štaviše automatski se pamla iri poslednja brisanja!

WordPerfect na ekranu na poravnava desnu marginu (na slempaču je poravnava ako korisnik ta želi). Izmana u okviru reda vida se samo u tom lalom redu i to samo dok je kursor u njemu. Ceo tekst se formatlzuje od početka dekumenta do položaja kursora. Se za aventualno ponovno formatizovanja treba otići na sam kraj teksta. Ovo je prilično naugodna osobina jer korisnik često za lranutak gubi kontrolu nad položajem kursora. Na primar pišeta nešto i hoćeta da dodate reč-dva u gornjem redu. Umatanjem se gornji red ašn često i preko desne ivice teksta. Vraltte kursor nadola da biste naslavili lamo gde ste stali — all *WordPerfect* nazavisno od vas prvo prelama red a samim tim i pasus a varovatno i čitav lekst na ekranu. Se tek onda spušta kursor tako da na možete odmah da nastavite sa pleenjem. U naknadu za te specifičnosti pralom ekrana se odmah vidi na ekranu kao horizontalna crta. To je izvanredna osobina verovatno ubačena pe ugledu na *Wordstar*.

Navedanom naporedu značajnih tastera može se prigovoriti da nlje argonomski jer primorava ruka da se odmaknu od tastatura. Tima se kucanja automatski usporava. Idealno bi bilo da svaki korisnik može da definiše sopstvenu tastaturu. *WordPerfect* 5.0 to omogućava datotekame.

Nabavka i cene

Program sa može poručiti direktno od proizvođača sa adrese: WordPerfect Corporation 1555 N. Technology Way, Oram Utah 84057 USA, telefon 801 801 225 5000, telex 820618 fax 861 801 227-4266. U tom slučaju košte nekih 450 dolara, što je čak i za američka pojmova mnogo novca sa jedan procesora reči. Prava cena je oko 200 dolara i to u radnjama za prodaju softvera na malo. Inače, WordPerfect Corporation ima 19 predstavnika na svim kontinentima. Dajamo neka adrese koja bi mogla biti interesantna za korisnika u Jugoslaviji: Francuska: WordPerfect France, 38 rue del Etete Generaux 78000 Versailles FRANCE tel. (181) 39 51 78 88. Grčka: Kronos Elaginonics S.A. 317 Mesogian Street 152 31 Athens GREECE tel. (01) 6029468. Velika Britanija: Sentinal Software Ltd. First Floor Wallington House New Zealand Avenue Walton on Thames Surrey KT12 1PY GREAT BRITAIN tel. (0932) 231164. Zapadna Nemačka, Austrija i Švajcarska: WordPerfect Software GmbH Frankfurter Straee 33-35 6236 Eschborn Wsel Germany tel. (06196) 48 17 22.

nja, kraj strane, razmak između linija i tako dalje. WordPerfect tu nije izuzetak, ali je veoma interesantan način na koji sam korisnik može da razgleda tekst iznutra. Alt-F3 otvara na ekranu dva prozora. Gornji je tekst u normalnom obliku, donji sadrži i tekst i vidljive interne simbole od kojih zavisi izgled dokumenta na ekranu i/ili štampaču. U oba ekrana kursor je na istom mestu, jedino donji prozor može po redovima biti prelomljen na sasvim drugim mestima u odnosu na „čist" tekst.

Korisnik može da menja tekst u donjem ekranu na uobičajeni način i ima da bitno utiče na izgled dokumenta. Opcija **Reveal Codes** jedna je od najjačih strana WordPerfect-a.

Na žalost, prisustvo posebnih internih kodova čini tekstove snimljene u standardnom formatu WordPerfect-a nečitljivim za druge programe. U verziji 5.0 je ubačeno mnogo novih kodova (ima ih tačno 115), tako da je neophodna posebna konverzija tekstova iz verzije 5.0 u prethodnu verziju 4.2. Sa druge strane, tekstovi iz verzije 4.2 bez problema se učitavaju u verziji 5.0.

Oblikovanje teksta

Prethodne verzije WordPerfect-a imale su probleme sa mešanjem slova različitih veličina u istom redu. Sada je taj problem otklonjan. Kombinacijom **Ctrl-F8** biraju se oblici slova po veličini (eksponenti, indeksi, sitna, mala, velika, vrlo velika i ekstra velika slova), odnosno po obliku (pojačana, podvučena, dvostruko podvučena, kurziv, naslov, osenčano, umanjena velika slova, ispravljen tekst i precrtana slova). Posebnom opcijom se svaki od ovih parametara može izdvojiti na ekranu, npr. da jedan oblik slova bude iste boje, pojačan itd., u zavisnosti od sposobnosti ekrana da sve to prikaže. Međutim, izbor oblika slova u tekstu i njihovo pojavljivanje na ekranu nemaju nikakve veze sa odštampanim tekstom. Oblik teksta na štampaču zavisi, naravno, samo od štampača.

Ako korisnik drukčije ne naznači, tekst će se štampati u standardnom obliku slova. Za matrične štampače obično je to Pica 10.

Parametri novog teksta menjaju se prostim biranjem oblika iz menija i tekst od tog mesta pa nadalje zadržava nove atribute. Ako se želi promeniti oblik već unesenog teksta, treba ga osvetliti kao blok i onda odabrati neku opciju iz Font menija.

Na ekranu se ipak vidi tačan prelom reda i strana. Na primer, red sa vrlo velikim slovima sadržaće možda svega 10—15 znakova. WordPerfect ne poravnava desnu ivicu reda na ekranu, pa tekst može izgledati veoma nepregledno, iako je precizna informacija o širini reda i dužini strane dragocena.

Kombinacija **Shift-F8** vodi u meni za definisanje parametara reda, strane i celog teksta. Što se redova tiče, moguće je definisati levu i desnu marginu kao rastojanja udesno i ulevo od ivice strane, zatim oblast u kojoj se vrši prelom reči na kraju reda (npr. 10% od trenutno važeće širine reda), da li red treba poravnati na desnoj ivici, razmak između redova, visinu reda i slično. Najvažnije opcije za formatizovanje strane su centriranje teksta vertikalno po strani, definisanje zaglavlja i podnožja stranice, vertikalne margine (rastojanje teksta od gornje i donje ivice papira), broj i položaj broja strane, veličinu i tip papira (A4, B5, specijalne veličine papira za advokate, koverte, proizvoljna veličina itd.). U ovoj opciji treba navesti da je širina papira 8 inča i visina 12 inča, jer je to standard za naše korisnike.

Opcije za ceo dokument i nisu tako brojne: veličina slova, početni oblik i vrsta slova, način upoređivanja novog i starog teksta (redline metod) i sumarni podaci o dokumentu (ko je pisao, ko je uneo, kojeg datuma i slično). Ostale opcije formatizovanja su raznorodne: definisanje tačnog položaja teksta u odnosu na ivice papira, uslovni kraj stranice, neke posebne funkcije štampača, kojim jezikom se piše (engleski, francuski — u okviru mogućnosti IBM-ove tastature da prikaže neka ne-engleska slova) i slično.

Posebne funkcije štampača su veoma korisne. Može se uključiti **kerning** — definisanje razmaka između pojedinih parova slova (veoma cenjena osobina programa za stono izdavaštvo). Takođe se može navesti niz ili datoteka upravljačkih znakova koji se šalju štampaču na početku štampanja — veoma značajno za naše korisnike ako nemaju hardverski definisana naša slova na štampaču. Može se navesti i tačan razmak između slova i reči, npr. da bude kao što je proizvođač štampača predvideo, ili da bude optimalan u smislu WordPerfect-a, ili da korisnik sam zada neku vrednost itd.

Konverzije tekstova

Program *Convert* omogućava konverziju tekstualnih datoteka u *WordPerfect* i iz *WordPerfect-a*. ASCII datoteke mogu se snimati i učitavati direktno iz samog programa, a *Convert* se koristi iz DOS-a. Zastupljeni su sledeći programi, odnosno formati: **Revisable-Form-Text** (most ka IBM-ovom DCA standardu za razmenu tekstova između računara i programa za obradu teksta), **Final-Form-Text** (pretvara **WordPerfect** u DCA), **Navy DIF** (standard mornarice SAD), **WordStar** (pretvara **WordPerfect** u WordStar), **MultiMate** (**WordPerfect** u MultiMate — nekada jedan od najpopularnijih procesora reči za velike firme), **Seven-Bit Transfer Format** (**WordPerfect** interno koristi osmi bit, slično *WordStar-u*, pa zbog slanja modemom treba snimiti samo sedam bitova svakog slova), **Mail Merge** (korisno za *dBASE*, *WordStar* i druge slične formate baza podataka, koristi se za slanje cirkularnih pisama), **WordPerfect Secondary Merge** (pretvara sekundarnu datoteku **WordPerfect-a** u oblik za dinamičku tabelu (spreadsheet)) i **Spreadsheet DIF** (obratno od prethodne opcije). Napomenimo da je skraćenica DIF u stvari prezime datoteka nastalih iz programa **VisiCalc**, koji se svojom popularnošću (prvi program prodat u preko 500000 primeraka) nametnuo kao standard za podatke iz dinamičkih tabela.

Stilovi formatizovanja

Stil formatizovanja je prosto imenovani skup parametara za oblik reda, strane i/ili dokumenta. Jednom izgrađeni stilovi se mogu po volji primenjivati na tekst, čime se štedi vreme, omogućava isti izgled dokumenata npr. u celoj radnoj organizaciji, ekspert za tehnički prelom teksta ne mora stalno da bude prisutan, i tako dalje. *WordPerfect* tek u ovoj, najnovijoj verziji ima stilove za razliku od *Word-a* i *Venture* koji ih imaju od samog početka. U tim programima stil je samo zabeležen niz parametara, dok je ovde stil i to i neka kombinacija teksta. Stil tako može poslužiti i kao neka vrsta skraćenice, npr. za naslove ili pozdravne poruke i slično.

Postoje dve vrste stilova: upareni (paired) i slobodni. Slobodni stil važi od položaja na kome je ubačen u tekst pa do kraja (ili do sledećeg stila). Upareni stil važi samo u bloku teksta, baš kao što i npr. pojačano (bold) slovo važe samo između dve markera [BOLD] i [bold].

Stilovi se lako i kreiraju i primenjuju na tekst. Za snimanje je dovoljno osvetliti blok teksta, pritisnuti **Alt-F8** i odabrati **Create** iz menija. Već snimljeni stilovi se pozivaju sa diska ili se biraju iz menija.

Stilovi su svakako dobrodošlo pojačanje u *WordPerfect-u*.

Prelom strane

WordPerfect nije program za crtanje, iako može da ubaci u tekst horizontalne i vertikalne linije. Pravi novitet u verziji 5.0 je slobodno mešanje grafike i teksta. Slika se može učitati direktno i biva smeštena u poseban okvir (box). Korisnik može prethodno da sam definiše okvir, odnosno da ga deklariše kao jedan od četiri predefinisane tipa: **Figure** (ilustracije), **Table** (tabele), **Text** (tekstovi) i **User-defined** (proizvoljna kombinacija prethodne tri tipa). Tip okvira ne utiče na sadržaj, već naznačava u kojoj od četiri predefinisane liste će se okvir voditi. Elementi svake liste numerišu se posebno i imaju sopstvena standardizovana obeležja. Naravno, preporučljivo je da se u okvir tipa **Figure** zaista smeštaju slike, u okvir tipa **Table** npr. statističke tabele, u okvir tipa **Text** citati, a da okviri tipa **User-defined** sadrže ono što se ne može smestiti u jednu od prethodne tri kategorije.

Tehnički detalji

Prikazujemo *WordPerfect* 5.0 za PC usaglašene računare (XT, AT, PS/2). Sistem treba da ima DOS i barem 384 K centralne memorije. WP.EXE koristi 236K, a ostatak je minimalna količina memorije potrebna za rad sa tekstom. Ako postoji proširena memorija (expanded memory) po standardu LIM, *WordPerfect* će je koristiti bilo za datoteke, bilo za sopstveno izvršavanje (pod uslovom da ima više od 300K raspoloživog prostora). Ništa naročito se ne dobija korišćenjem običnih RAM diskova.

Ovo je prva verzija *WordPerfect-a* koja radi sa grafikom, tako da su podržani samo najvažniji monitori i grafičke kartice. Monitori su monohromatski, CGA, EGA, VGA i PC3270, a program sam prepoznaje grafičku karticu. Na raspolaganju su: adapter bez grafike, Hercules 720×348 mono, Hercules InColor 720×348 u 16 boja, CGA 640×200 mono, EGA 640×350 mono, 320×200 u 16 boja i 640×350 u 4 i 16 boja, VGA 640×480 mono i u 16 boje, 320×200 u 256 boja, AT&T 6300 i Olivetti 640×400 mono i Compaq portabl plazma ekran 640×400 mono. Grafička kartica nije neophodna za rad sa grafičkim prikazom teksta, ali bi rad bez nje svakako bio otežan.

WordPerfect 5.0 radi na sistemima sa dve disketne jedinice, odnosno, sa tvrdim diskom. Sistemi sa dve disketne jedinice iznova postaju važni, jer cena prenosnih PC računara i dalje opadaju.

Preporučljivo je imati barem 512K centralne memorije. Međutim, dužina obrađivanog teksta ograničena je samo kapacitetom spoljašnje memorije.

Podržano je 123 štampača, a program *PTR* omogućava da se naprave novi drajveri. Nikakva uputstva nisu data za *PTR*, tako da korisnik mora da se snalazi pomoću HELP-a. Obično programi te vrste pitaju korisnika za kodove koje treba upućivati na štampač, a korisniku je dovoljno uputstvo za štampač. *PTR* može da se koristi i na taj način, ali neke opcije jednostavno nisu dokumentovane. Listanjem raznih drajvera ustanovljava se da postoje varijable, stringovi, a na jednom mestu primećuje se i IF naredba. Celina se ne vidi, pa ako hoćete ozbiljno da koristite *WordPerfect* 5.0, treba da imate štampač nekog od poznatih proizvođača, npr. IBM, Star, Apple, Panasonic, Epson i HP.

Za profesionalne korisnike važna je činjenica da *WordPerfect* 5.0 podržava i HP LaserJet i Postscript jezike, a da je izbor kertridža i softverski definisanih vrsta slova (fontova) moguć direktno iz programa. Vrste slova mogu se menjati i u toku štampanja, naravno, ako sam štampač to podržava. Korisnik bi (teoretski) mogao da definiše i nove vrste slova, jer je širina slova sastavni deo drajvera za štampače.

U tekstu se vidi samo prazan imenovani okvir, ali tačnih dimenzija u odnosu na stranicu i ostatak teksta. Tekst može da kruži oko okvira što WordPerfect približava programima za stono izdavaštvo.

Još jedna bitna novina u ovoj verziji je mogućnost pregleda slike i cele strane štampanja. Nad uvezenom slikom mogu se vršiti sledeće operacije: **Move** (slika se pomera horizontalno ili vertikalno po okviru), **Scale** (umanjenje ili uvećanje slike), **Rotate** (rotiranje za zadati broj stepeni; takođe je moguće vertikalno zrcaljenje, tj. simetrično preslikavanje leve strane u desnu i obratno u odnosu na y-osu slike), **Switch** (invertovanje belo u crno i crno u belo; i može se primeniti samo na slike koje su nastale u tzv. bit-image programima), **% Change** (procenat transformacije koja se izvodi tokom premeštanja, rotiranja i menjanja veličine slike).

S obzirom da se svaki okvir vodi pod rednim brojem u posebnoj listi, može se direktno doći do željene slike, što znatno povećava komfor u radu.

Za svaki okvir može se posebno namestiti vrsta linije kojom će se iscrtati okvir, spoljašnji i unutrašnji razmak teksta i slike od okvira, položaj i broj naslova slike itd.

Mešanje slike i teksta do sada je našim korisnicima uglavnom bilo dostupno samo kroz program *Ventura Publisher*. *WordPerfect* je u tom pogledu tek nešto slabiji od *Ventura*, a trenutno je sigurno bolji od *MS Word-a* (u kome se slika može samo odštampati sa diska).

Konac delo krasi

Na žalost štampanje je pravljeno po ugledu na DOS naredbu PRINT. Sve što treba odštampati šalje se u red čekanja i svaki dokument dobija svoj redni broj. Samo štampanje počinje tek kada se to eksplicitno naredi. Ostale opcije su uobičajene: štampanje samo jedne strane, od strane do strane, kvalitet štampanja itd. Izgled dokumenta bitno zavisi od odabranog štampača, pa ako promenite drajver za štampač, WordPerfect će automatski preformatirati ceo dokument. Drugim rečima, moguća su iznenađenja i teško bi, na primer, bilo štampati probne primerke rukopisa na matričnom, a zatim bez izmena uraditi isto to na laserskom štampaču.

Može se birati između srednjeg (bržeg) i dobrog (sporijeg) kvaliteta štampe za slike.

Kao što je spomenuto, tekst se može u grafičkom režimu pregledati na ekranu. Postoje tri veličine kompletne stranice: 100% uvećana, 200% uvećana, a tu je i opcija da se pregleda susedna strana (facing pages). Kompletna stranica je nečitljiva, ali se u njoj vidi tačan položaj teksta i grafike. Uvećana slika stranice pokazuje jedan potencijalni problem za naše korisnike: *WordPerfect* predviđa samo standardna IBM-ova slova (svih 255), pa ako imate hardverski ugrađena naša slova ili ako želite bilo kakve posebne znake — to se neće videti u grafičkoj imitaciji štampanja. Bilo kako bilo, grafika je tu i značajno će unaprediti npr. izradu tehničke dokumentacije i sličnih tekstova.

WordPerfect sada liči na program za stono izdavaštvo — ali to ipak nije. Osnovni nedostatak je što se ništa ne može menjati dok je tekst prikazan grafički. I izbor vrste i oblika slova zavisi od štampača, a program ih ne generiše sam — prosto ih koristi. Ali, *WordPerfect* 5.0 je svakako korak ka stonom izdavaštvu i pokazuje u kojem pravcu će se današnji programi za obradu teksta razvijati.

Komunikacija sa diskom

Ova oblast je oduvek bila jedna od jakih strana ovog programa. Taster F5 vodi u meni za rad sa datotekama. Na ekranu se prikazuju dva stupca sa parametrima datoteka, kursor označava na koju se datoteku odnosi odabrana radnja iz menija: **Retrieve** (učitavanje), **Delete** (brisanje), **Move/Rename** (premeštanje i preimenovanje), **Print** (štampanje), **Text In** (učitava ASCII datoteku), **Look** (razgleda sadržaj datoteke bez učitavanja), **Other directory** (menja imenik), **Copy** (kopira datoteku), **Word search** (ispituje da li postoji navedena reč u datoteci), **Name search** (korisnik unosi početna slova a kursor se automatski namešta na datoteku). Sve u svemu, učitavanje datoteka je dobro (iako ne i sasvim idealno) rešeno.

Snimanje tekstova je pomalo zamršeno. *WordPerfect* stalno pita da li da presnimi tekst preko postojeće datoteke istog imena. Na sreću, sasvim je lako napraviti makro naredbu koja snima „sama od sebe". Može se narediti i automatsko snimanje na određeni period vremena, što je bitno za profesionalnu primenu.

Snimanje je prilično brzo, ali postoji i opcija **Fast Save** za snimanje u internom zapisu. No takva datoteka je sasvim nečitljiva za bilo koji drugi program osim *WordPerfect-a*. Uzgred, ni obične datoteke nisu ništa bolje: pune su specijalnih znakova za formatizovanje.

Grafički programi

WordPerfect 5.0 sarađuje sa [illegible] jim grafičkim programima. Postoje [illegible] na da se slika ubaci u *Word*[illegible] (1) programom *GRAB* i (2) direktnim [illegible] slike sa diska. *GRAB* se učita kao [illegible] (memory resident) program i pritiskom na kombinaciju Alt-Shift-F9 ekran biva uhvaćen i snimljen kao datoteka sa karakteris[illegible] WPG. GRAB zauzima manje [illegible] 15K i radi sa *AutoCAD-om*, *Framework*[illegible] ostalim važnijim programima.

Uvoz slike sa diska je jednostavan [illegible] su sledeći formati: CGM (*Computer Graphics Metafile*), DHP (*Dr Halo PIC* datoteke), DXF (*AutoCAD*), EPS (*PostScript* [illegible] jezik za laserske štampače), HPGL (standardni jezik za plotere), IMG (*GEM Paint*), MSP (*Microsoft Windows Paint*), PCX (*PC Paintbrush*), PIC (*Lotus 1—2—3*), PNTG (*Macintosh Paint*), PPIC (*PC Paint Plus*), TIFF (standard za razmenu grafike) [illegible] WPG (interni zapis slike u samom *WordPerfect* u) Kvalitet slike je bolji ako se upotrebljava neki od ovih formata nego ako se koristi GRAB.

Instalacija

Korisnik može ručno da [illegible] potrebne datoteke na tvrdi disk, [illegible] upotrebiti program INSTALL.BAT [illegible] *Learning*. INSTALL otvara imenik [illegible] na tvrdom disku kao i podimenik /LEARN za [illegible] primere raznih datoteka. Po završenom kopiranju na disk korisnik mora prvo da [illegible] pomoćnog programa odabere štampač [illegible] dokumenta na ekranu zavisi od osob[illegible] štampača. Program za instalaciju umeće [illegible] FILES-20 u CONFIG.SYS datoteku, [illegible] argument FILES naredbe već ne premašuje [illegible].

Instalacija je dakle jednostavna. *WordPerfect* 5.0 zauzima nešto više od 1.5 megabajta prostora na tvrdom disku. Pri tome INSTALL ne kopira neke potencijalno [illegible] programe, npr. *PTR.EXE* koji služi za instaliranje novih drajvera za štampače ili *CONVERT* za konverziju datoteka iz i u *WordPerfect*.

Postoji i poseban program za učenje *WordPerfect-a*. Aktivira se prostim TUTOR iz DOS-a, s tim da se prethodno mora preći u podimenik LEARN. Iskusni korisnici mogu obrisati sve datoteke za učenje kao i [illegible]

Pokretanje programa

WordPerfect se aktivira naredbom WP iz DOS-a. Mogući su i neki parametri [illegible] slika WP/B-30, što naređuje *WordPerfect*[illegible] snima tekst na disk svakih 30 minuta. Važnije opcije su: D (disk i/ili imenik u [illegible] se nalaziti radne datoteke), ime-datoteke ([illegible] navedena datoteka će se učitati u *Word*[illegible]), I (položaj pomoćne datoteke WP FIL — [illegible] opcija se retko koristi), M-ime-makro-[illegible] (makro naredba koja će se izvršiti [illegible] posle učitavanja programa) i [illegible]. Često korišćena kombinacija može [illegible] SET (privremeno) postaviti u [illegible] područje DOS-a ili izvršavati kao [illegible].

Po učitavanju korisnik vidi potpuno [illegible] zan ekran, ako ne računamo najniži [illegible]. Kroz njega se odvija komunikacija sa korisnikom. Tokom pisanja tu se pokazuje [illegible] datoteke, da li je prvi ili drugi doku[illegible] (*WordPerfect* može da radi sa svega [illegible] dokumenta istovremeno), broj strane, [illegible] na strani i položaj kursora u okviru linije. [illegible] kakvi meniji se ne ukazuju, lako da se korisnik oseće sasvim bespomoćnim.

[illegible] druge strane, iskusni korisnici najviše vole prazan ekran da bi odjednom videli [illegible] teksta. Sve je to u skladu sa ciljem [illegible] je ovaj program nasta[illegible]

Ne treba naročito isticati da postoji mogućnost stapanja tekstova (merge), npr. zajedničko štampanje poglavlja knjige, a takođe i da postoji opcija za stvaranje cirkularnih pisama (mail merge). Slogovi adresa se u programu označavaju posebnim kodovima, npr. **Ctrl-E** za kraj sloga i slično.

Za svakog po nešto

Bilo bi stvarno čudno kad vodeći svetski program za obradu teksta ne bi nudio i sortiranje, unos podataka i štampanje formulara, matematičke operacije (čitave formule — ne samo četiri osnovne operacije), posebne pogodnosti za pravničku struku, štampanje naslepnica, razne mogućnosti numerisanja pasusa, štampanje stranice sa uvlačenjem teksta radi olakšanog povezivanja, skiciranje sadržaja (outline processing), razgledanje predloga izmena teksta (red-lining), komentare u tekstu, stupce (do 24 na jednoj strani, bilo novinske, bilo obične paralelne), tezaurus (ne samo rečnik sinonima nego i antonima), proveru spelovanja, brojanje reči u tekstu, pretraživanje i promene, tabulatore, pristup dokumentima samo preko lozinke, generisanje proizvoljnih spiskova, automatizovano formiranje sadržaja i indeksa dokumenta, spiska ilustracija, referenci i tako dalje.

Zaista, *WordPerfect* je godinama usavršavan i predstavlja veoma kompletan program za obradu teksta. U tom smislu često se čuje pitanje: da li je *WordPerfect* najbolji program na svetu? Odgovor je svakako NE i to iz dva razloga: sama reč najbolji dopušta veoma različita tumačenja, a osim toga *WordPerfect* ni slučajno nije jedini dobar program na tržištu. U njemu ima oko 260 naredbi, a novi program *Wordstar 2000* ver. 3.0 ima tačno 662 mogućnosti. Pitanje je zapravo pogrešno postavljeno. Možda je trebalo da glasi: da li ću biti u stanju da uradim svoj posao (pišem, prekucavam, prevodim, dopisujem se) koristeći *WordPerfect*? Odgovor je naravno DA!

Duško Savić

Računar sa hiljadu lica

Jelo u tuđem tanjiru je uvek slađe, a program na tuđem računaru uvek radi bolje. Nijedan računar ne može da oponaša toliko euštinski različitih mašina kao „Tramijalov kameleon". Emulator je produkt pomoću koga vaš kompjuter funkcioniše kao potpuno druge mašine. Od ovih veoma ljudskih slabosti možda najviše pate vlasnici „eteri ST" računara. Ovaj članak objašnjava šta su emulatori, koje su njihove mogućnosti, kako rade i na koje probleme možete naići pri njihovom korišćenju.

Da li se ponekad osećate kao da su svi dobri poslovni programi napisani upravo za svaki drugi računar sem za vaš? Do sada rešenja su bila potpuno neprihvatljiva — ili da čekate da se taj program napiše za vaš računar da prodate veljenog „starija i kupite odgovarajuću konfiguraciju drugog računara ili da nabavite ST klon tog programa. U prvom slučaju trebalo bi vam bar dva života dok dočekate da se iz programa otklone sve bagovi drugo rešenje automatski otpada a u trećem bi bilo moguća da se vaš novi dBASE 2 klon ponaša više kao dBASE 1 1/2. O tome da se na brzišlu u međuvremenu pojavio već dBASE IV i da ne govorimo. Danovna prednost emulatora je baš u tome što omogućava korišćenje programa sa drugih računara koji u originalu nisu napisani za vašu mašinu.

Trik broj 1 Kompatibilnost

Postoji dosta prepreka na putu da jedan kompjuter emulira drugi. Bilo bi zaista lepo kada bi se jednostavnim slavljenjem makovog diska u ST-ov drajv mogao startovati makov program. Na kraju makovi diskovi su fizički identični starijevim: imaju 80 traka i 135 TPI. Obe mašine imaju istu Moforolu 68000 i približno istu memoriju. U čemu je problem? Postoji puno tehničkih problema koji odražuju nivo kompatibilnosti.

Trik broj 2 Ako ne postoji emulator

Trenutno postoji veoma mali broj emulatora kako za stari ST tako i za ostale računare. Ako želite da koristite softver pisan za XYZ kompjuter a na postoji XYZ emulator možete samo da očajavate. Na primer trenutno na postoji način da „amigini programi rade na „starija

Trik broj 3: Softverske kompatibilnost

Treći trik je u tome da ako jedan program napišen za kompjuter X radi na kompjuteru Y ne znači da će raditi i ostali. Ovo je veliki problem kod svih emulatora, jer je veoma teško napraviti savršeno simuliranje jednog kompjutera na drugom. Recimo da sa neka rutina na datom kompjuteru koristi samo jednom od sto puta. Ako u emulator ugradimo ovu rutinu možemo uštedeti sve programe za 50%! Kod emulatora brzina se uvek mora žrtvovati ali većina programera će odlučiti da na ugradi retku rutinu u emulator ako ga ona uštede usporava.

Trik broj 4 Brzina

Sledeći trik je u tome da većina programa koji rade pod emulatorom rade mnogo sporije od očekivanog. Motorola 68000 u ST-u radi na 8 MHz što odgovara brzini od osam miliona ciklusa u sekundi. 8 MHz je veoma velika brzina dovoljna za rotaciju solidnih 3D objekata i brzo skrolovanje ekrana pa ako vidite program koji sporo radi možete biti sigurni da programer nije u potpunosti iskoristio potencijal procesora. Kada se kaže da je brzina nekog procesora 4.77 MHz misli se na broj taktnih ciklusa izvršavanja jedne mašinske instrukcije. Obično zahteve od 4 do 30 tih jednostavnih operacija. Iako ST radi na 8 MHz potrebno mu je mnogo više od 0.125 miliontog dela sekunde da izvrši jednu kompletnu mašinsku instrukciju. Veoma je važno koliko je ciklusa potrebno kompjuteru za izvršavanja jedne instrukcije. Dok komodore 64 radi na 1.5 MHz i na izvršavanja jedne instrukcije troši 2—3 ciklusa. IBM PC XT koji ima klok od 4.77 MHz treći za jednu instrukciju od 10 do 20 ciklusa. Na kraju ispada da IBM i C64 imaju istu brzinu pri matematičkim operacijama. Iako prvi ima tri puta brži klok. Zato je brzina emulatora vrlo relativna stvar. Veoma je važno znati programirati emulator da kaže „Aha znam šta nekoliko sledećih 6088 instrukcija rade. Umesto da ih polako emuliram jednu po jednu izvršiću ih sve zajedno u jednoj MC 68000 instrukciji.

Trik broj 5. Kompatibilnost disketa

Za emulaciju je vrlo bitno da računari koriste isti format diskete. Kako bi pročitali podatke se 5 1/4 diskete ili makove diskete na „stariju ST ? Atarijevi drajvovi čitaju IBM-format dvostruka gustina. Za sve ostalo je potreban dodatni hardver.

Postoje tri načina da te podatke pročitate.

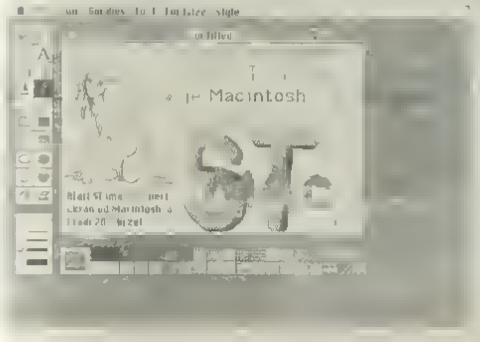
Prvi je eksterni 5 1/4 inčni disk-drajv koji možete nabaviti bilo gde u inostranstvu. Postoje dva modela: Paradox Microbyte i IB Computers drive i koštaju oko 250 USD.

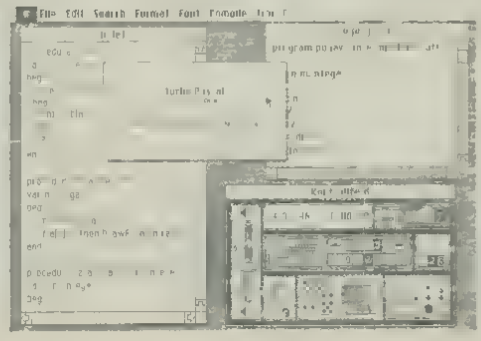
Drugi je Data Pacific Translator One koji se koristi sa Magic Sac emulatorom za čitanje „makovih 3 1/2 inčnih disketa.

Treći najjeftiniji (ali najfleksibilniji) način je da fizički povežete dva kompjutera kablom i prenesete podatke sa jednog na drugi. Pri ovom postupku moguć je veliki broj grešaka i treba ga primenjivati samo u krajnjem slučaju. Treba misliti i na zaštitu jer u slučaju da su programi zaštićeni možete da zaboravite bilo kakvu emulaciju.

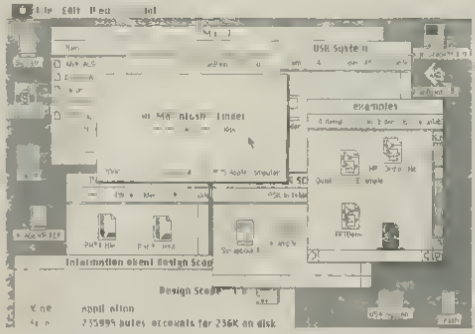
Trik broj 6: Hardverski emulatori

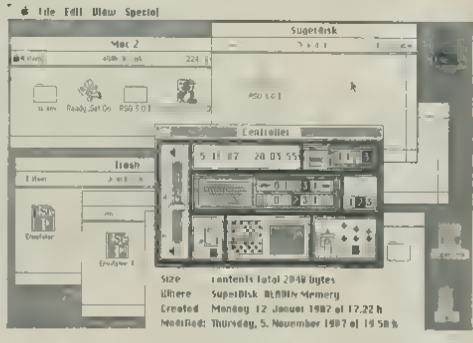
U poslednje vreme se težišta pranosi na hardverska emulatora. Ideja se sastoji od dodа-





Mekintoš na „atariju ST





vanje novog procesora ST-u. Procesor 8088 izvršava 8085 instrukcije veoma brzo i obezbeđuje dobru brzinu emulatora. Iako je Atari najavio IBM hardverski emulator još davne 1986 godine do dana današnjeg se nije pojavio ni njegov ni bilo koji drugi. Prvi bi trebao da bude Paradox-ov Parabox zasnovan oko 10 MHz NEC V20 koji bi radio do 30% brže od IBM XT-a. Treba da se pojavi krajem godine. Još jedna mogućnost je korišćenje „starija" kao terminala za IBM procesor. IBM PC kompatibilne ploče se mogu naći veoma jeftino, jer veliki deo cene odlazi na monitor, tastaturu i napajanje.

Zašto su spori?

Četiri najpopularnije mikroprocesora danas su 6502 (Atari 130 XE, C64, epl 2, BBC B), Z80 i 8080 (spektrum, amstrad CPC, CP/M), 8086 i njegov rođak 8088 (IBM PC i kompatibilci) i MC 68000 (Atari ST, Apple Mac, Amiga). Svaki od ovih procesora razume kompletno drugačiji mašinski jezik koji je u stvari niz kodiranih brojeva. 8080 na primer koristi 243 da označi JUMP instrukciju sličnu bejzik naredbi GOTO, dok 68000 koristi 78 250. Pri emulaciji makovog 68000 programa na starijevom 68000 poseo emulatora je da prevede neke manje razlike između Mac i ST-a a eko umesto Aladina koristila Magic Sac koji ima ugrađene makove ROMove. Ataru ne mora da troši svoju brzinu čak ni na njihova emuliranja. Sa druge strane, ST Transformer koji emulira „stari" 130 XE na ST-u mora da prevodi 6502 kodove u 68000 instrukcije. Pri emulaciji PC-ja ST ne razume ni jedan broj koji mu program šalje pa mora da ih prevodi u odgovarajući MC68000 — jedan po jedan. Za svaku emuliranu instrukciju program mora da

— nađe tranutni operativni kod (kod sledeće 8088 instrukcije)
— uveća adresni pointer tranutne instrukcije
— pogleda ošeau rutine koja emulira pribavljeni kod
— skoči na tu rutinu
— izvrši je
— skoči nazad u kod da nađe sledeći operativni kod i tako dalje

Ovaj postupak se ponavlja za svaku 8088 instrukciju što znači da je za izvršavanje jedne 8088 instrukcije potrebno oko 8 MC 68000 instrukcije. To objašnjava zašto pc-Ditto ima brzinu od 1 MHz. Emulator dakle funkcioniše kao interpretar instrukcija procesora koji se oponaša. Rešenje za ovaj problem bi bilo da Motorola 68000 na početku programa prevede sve 8088 instrukcije u svoj kod pa da ga posle izvršava. Iako je ovo teoriski moguće postavlja se problem „skokova" u programu koji bi trebalo da se re-kodiraju a i ne bi radili programi koji menjaju sami sebe u izvršavanju.

Na kraju postavlja se i pitanje

CP/M Z80

Brzina 1 MHz
Softverska kompatibilnost: visoka
Kompatibilnost čiskete nekompatibilne
Vrsta emulatora: Softverski

ST Transformer

Brzina 1 MHz
Softverska kompatibilnost visoka
Kompatibilnost čiskete nekompatibilne
Vrsta emulatora Softverski

pcDitto

Brzina 1 MHz (60% brzine XT-a ili 30% brzine Turbo XI-a)
Softverska kompatibilnost: visoka
Kompatibilnost čiskete nekompatibilne
Vrsta emulatora: softverski

Aladin 1.3

Brzina: 8 MHz
Softverska kompatibilnost visoka
Kompatibilnost čiskete nekompatibilne
Kamentar: Dosta programa koriste Atarijev povećani displej od 640×400
Vrsta emulatora Softverski

Parabox IBM emulator

Brzina do 30% brži od IBM-a — 10 MHz
Procesor NEC V20
Vrsta emulatora Hardverski

Magic Sact

Brzina 8 MHz
Softverska kompatibilnost rada svi programi koji ne koriste nova ROM-ova
Kompatibilnost diskete nekompatibilne ali je moguće čitati makove diskete sa Translatorom
Vrsta emulatora Hardverski

Šta očekivati?

Ako na igrate arkadne igre, mislim da vam brzina emulatora neće predstavljati problem. Razmislite koliko često će vam biti potrebne punih 8 MHz? Većinu svog vremena procesor lenjo čeka da se nešto desi: da pritisnete taster ili pomerite miša, otvorite datoteku itd. Ako vas interesuje obrada teksta ne verujemo da ćete primetiti razliku da li procesor čeka milion ili osam miliona puta u sekundi. Da prijsnete taster ali ako se bavite kompjuterskom animacijom razlika će već biti očigledna.

Pored velikog broja svojih programa koji rade punom brzinom, stari ST može da koristi i programe za mekintoš koji pomoću „Aladina" rade 20% brže nego na maku (osim s diskom) a i ekran je veći za 17%. Tu su još i programi PC-ja koji isto znatno sporije rade. Na prošlom Sajmu tehnike imali smo priliku da se sa hard diska kompjutera PS/2—60 prenesemo oko 10 M programa na diskete a zatim ih direktno učitamo u starije i svi su radili do jednog!

CP/M Z80 emulator

Najveći problem ovog emulatora je — brzina. Brzine od 1.0 MHz nije baš toliko mala ali pri poređenju se nekim novijim CP/M mašinama kao što su ATR-8000 ili mikromini SB180 izgleda smešno. Za one koji su navikli da rade na 2.0 MHz CP/M kompjuterima ovo neće predstavljati nikakav problem. Kompatibilnost je dosta visoka: svi programi rade besprekorno. Drugo stranu medalje predstavljaju diskete — 3 1/2 inčne CP/M diskete nisu kompatibilne sa atarijevim formatom a slera diskete bilo u 5 1/4 Kaypro formatu ili 8 inčne teško se mogu pročitati i pomoću eksternog drajva. Emulator nije baš jednostavno podesiti ali kad se podešavanje završi stari predstavlja sasvim prihvatljiv CP/M sistem.

ST Transformer

Ovo je primer dobro urađenog emulatora — iako se setimo da stari 800 radi na 1.5 MHz brzina koju ovaj emulator postiže je izvanredna — 1 MHz! Softverska kompatibilnost je iznenađujuće dobra a obzirom da prva verzija ovog emulatora nisu mogle da pokrenu ni običan bejzik. Velika većina nezaštićenih programa će raditi — kao i popularni *Paper Clip* procesor. Iskste iako igre mogu biti malo spore. Moguća je da će nako nekad napraviti i disk koji bi čitave double density stari 8-bit diskete sa eksternog 5 1/4 drajva. Slične se utisak da ovaj program nije napisan za obične korisnika već za stare atarijevce.

Parabox IBM

O ovom emulatoru se ne zna puno osim da se ulaže na naki od portove ST-a i da radi brže od originalnog IBM PC XT-a. Približna cena (za sva zainteresovane) je 350 USD ▶

pcDITTO

Autor ovog programa Bil Tal (Bill Teal) (Avant-Garde Systems) zaslužuje svaku pohvalu za ovaj izuzetan emulator. Pre *pcDitta* svi „atarija" i IBM-a su se ponašali kao ulje i voda, jedan pored i oko drugog ali bez mešanja. I dok je bilo moguće preneti datoteke sa jednog na drugi, programi su bili potpuno druga stvar. Programi pisani za PC-DOS i MS-DOS jednostavno nisu mogli da rade na Motoroli 68000. Bil Tal koji je inače radio u IBM-u proveo je 18 godina od kojih je mnoge posvetio pisanju emulatora u kompjuterskoj industriji. Za čoveka koji može da „natera" borouz da se ponaša kao IBM 370 pisanje IBM emulatora za „atari ST" je bila prava šala.

Trebalo je samo razraditi neka detalje — njih par stotina hiljada, zbog kojih program može da padne.

Program *pcDitto* je potpuno softverski emulator koji pretvara svaku 8088 instrukciju u jednu ili više MC 680000 instrukcija. Može da izvršava programe sa 60% brzine originalnog XT-a koji radi na 4.77 MHz ili sa 30% brzine turbo XT-a sa klokom od 8 MHz. Praktično, to znači da na njemu bolje rade poslovni programi (baze podataka, spreadsheets) nego arkadne igre ili drugi programi koji imaju funkcije za rad u realnom vremenu.

Radi i na 520 ST i na 1040 ST, dugačak je oko 150K, ali ipak ostavlja dosta mesta za programe. Na 520 ST ima 364K slobodne memorije, dok na 1040 ST ima čak 703K, što je više nego na maksimalno proširenom 640K IBM PC-u! Najnovija verzija programa radi kako na kolor tako i na monohromatskom monitoru i može da imitira bilo monohromatski bilo kolor grafički adapter. Podržani su paralelni i serijski portovi kao i 3 1/2 inčne diskete. Dok „atari" može da čita diskete formatirane na IBM-u, PC ima problema sa „atari" formatiranom disketom, ali ako ste vlasnik „atarija", to vam verovatno uopšte nije ni bitno.

Startovanje sistema može biti pomalo konfuzno za korisnika naviknutog na prednosti GEM-a. Ono se može obaviti kako sa diska tako i sa hard diska. Problemi počinju ako koristite eksterni 5 1/4 floppy disk. Pošto se program mora startovati sa 3 1/2 drajva, sistem pretpostavlja da je eksterni drajv drajv A jer se sa njega butuje, pa traži da u drajv A (fizički drajv B) stavite sistemski disk. Od tog trenutka pa nadalje fizički drajv B je logički drajv A, a fizički drajv A je logički drajv B. Jasno?

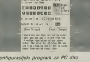
Stvar se naročito komplikuje korišćenjem „pametnog" IBMovog softvera koji automatski prepoznaje konfiguraciju sistema i očekuje da ćete tu konfiguraciju i koristiti! Ako želite da startujete program i koristite ga kao da imate samo jedan drajv da ne biste prebacivali podatke sa 5 1/4 inča na 3 1/2, program će reći „Aha, on ima dva flopija" i koristiće drajv B za podatke, a vama neće ostati drugo osim da prebacujete fajlove sa A na B. Nivo kompatibilnosti je visok i u mnogome zavisi od DOS-a koji koristite. Može se reći da sigurno rade PC-DOS 1.0, 1.1, 2.0, 2.1, 3.0, 3.1 i 3.2, a za sve ostale verzije nema garancije da će potpuno raditi.

Probali smo veliki broj programa uključujući i *Lotus 1-2-3, WordPerfect, SideKick, PCTools, Mash CAD i GW Basic* i nismo imali nikakvih problema. Jedino nismo mogli da probamo zaštićene igre koje zaobilaze DOS i startuju iz reseta.

Sa pc Ditto-m se dobija i program *PC-menu* koji omogućava promenu karakteristika sistema kao što su grafički adapter, startne boje ekrana, disk drajvovi i tastatura. Jedini hardver koji ne podržava *pcDitto* je miš, ali iz Avant Garde Systems-a obećavaju da će sledeće verzije programa imati ugrađenu i tu opciju. Sve u svemu, bili smo impresionirani *pcDitto*-m iako smo navikli na sve prednosti GEM-a. Ako vam smeta to što na PC-ju nema propadajućih menija, ikona i dialog boksova — jednostavno učitajte GEM!

MAGIC SAC+

Zbog Appleove filozofije da programi ne treba da budu zavisni od hardvera, 90% „makintoš"



Konfiguracijski program za PC dito



Startni ekran programa PC dito



Emulator CP/M-Z80

programa rade sa ovim emulatorom i to često brže nego na originalnom „maku"! *Magic Sac+* se sastoji od kertridža koji se utakne u ST-ov kertridž port, boot diska u ST formatu, diska za prenos softvera u „mak" formatu i kabla za povezivanja ST-a i „maka".

Startovanje ovog emulatora je veoma zanimljiv događaj. Pre svega, potrebna su vam dva „makintoš" 64K bit ROM-a označena sa 342-0220-A i 342-0221-A ili 342-0220-B i 342-0221-B. Moraju se koristiti originalni ROM-ovi jer EPROM-i neće da rade. Ovi ROM-ovi se mogu nabaviti kod bilo kojeg ovlašćenog servisera proizvoda firme Apple-a ili u radnjama koje prodaju *Magic Sac+*. Ako za to nemate mogućnosti, jedino vam ostaje da kanibalizujete „makintoš".

I pored toga što su diskovi fizički identični, „mak" ne može da čita atarijeve diskete kao i obratno. Da biste mogli da koristite Apple-ova programe morate da imate i jednu stvar čim koje nije u sklopu *Magic Sac*-a — pravi „makintoš".

Prebacivanje softvera na „atari" počinje formatiranjem disketa *MacFormat* programom, a završava prebacivanjem programa pomoću kabla i softvera koji se dobijaju uz *Magic Sac*. Na žalost, kabl je potpuno neupotrebljiv kod „maka plus" i SE-a, pa bi u tom slučaju morali da nabavite Data Pacific-ov *Translator* koji se povezuje sa atarijem preko porta za eksterni drajv i MIDI-ja a omogućava direktno čitanje „makovih" disketa. Ovo zadovoljstvo bi vas koštalo 279.95 USD.

Veliki broj programa — *Ready Set Go, Comic Works, Mac Draw, Cricket Graph, Turbo Pascal, Page Maker, Super Paint, Mac Project*, Reflex i Microsoft Word — rade bez problema i koriste atarijev povećani displej od 640×400 tačkica za razliku od onih koji koriste ilegalne pozive operativnog sistema ili rade sa standardnih 512×342, a to su uglavnom Apple-ovi programi i programi iz javnog vlasništva. Najočigledniji primeri su komunikacijski i MIDI programi. Da bi ispunili uslove tajminga, programeri se često obraćaju direktno specifičnom hardveru mašine kao što je „raskov" serijski čip. Normalno, kada program potraži taj čip na ST-u, mašina se blokira. Čak i ljudi iz Apple-a veruju da neka od njihovih pravila postoje da bi se kršila — možda baš zato *MacWrite 4.5* ne radi na „maku 2".

Vlasnici 520 ST-a mogu da simuliraju 128 K ili 256K „maka", što je nedovoljno za većinu programa. Oni koji imaju 1M mogu da konfigurišu njihova *Magic Sac*-ove kao „makintoše" 832 ili 512 K, s tim što bi ovi drugi ostatak memorije koristili kao ram-disk.

Ako imate kolor monitor, možete koristiti *Magic Sac*, ali ta kombinacija ne daje kolor „makintoš", a takođe ni veliki izbor boja jer ih zbog kontrasta morate podesiti na crno-belo. Za iole ozbiljniji rad potreban vam je monohromatski monitor iz Atarijeve serije SM.

„Makintoš" ima dva serijska porta, jedan za komunikacije a drugi za printer. Sa druge strane, ST poseduje serijski port za komunikacije i paralelni za printer. Pošto većina Mac softvera zahteva *Imagewriter*, moraćete da instalirate sopstveni drajver koji podržava vaš printer.

Ako niste ranije radili na „maku", biće vam potrebno malo vremena da se naviknete na novi korisnički interfejs. Kod ST-a, „maka" i „amige" postoje male razlike. To je urađeno da bi svaki model bio jedinstven i zaštitio kompaniju od suđenja.

Mali problem vam može predstavljati to što „mak" koristi *Pulldown*, a ST *Dropdown* menije. Razlika i nije tako drastična — samo se morate navići da držite levi taster pritisnut dok šetate kroz meni.

Makova tastatura ima dva tastera <COMMAND> i <OPTION> koji ne postoje na ST-u, ali se oni simuliraju ST-ovim <CONTROL> i <ALTERNATE>. Takođe postoji i razlika između izbacivanja disketa — dok je „mak" izbacuje automatski, kod ST-a se to mora raditi ručno. Zbog toga *Magic Sac* ispisuje A ili B u gornjem desnom uglu ekrana kad treba izbaciti disk.

Vlasnici ST-a sa jednim drajvom će morati da se igraju „disk-džokeja", dok eventualno, kao i svi „makovi" korisnici, ne uvide da je nemoguće raditi bez drugog drajva. Ovaj emulator podržava i hard disk.

ALADIN

Aladin je za razliku od *Magic Sac*-a potpuno softverski emulator „makintoša" i radi samo sa monohromatskim monitorom. Brzina i softverska kompatibilnost su iste kao i kod *Magic Sac*-a, a može se konfigurisati na bilo koju količinu memorije od 128 K do 2 M. Zauzima jednu jednostranu disketu i zaštićen je (iako se lako kopira *ProCopy*-jem 4.01). Ima ugrađen ram-disk čija veličina zavisi od slobodne memorije. Uz njega se dobijaju drajveri za *ImageWriter* i EPSON kompatibilne štampače. Pri resetu, koji se može obaviti pritiskanjem reset tastera ili <CONTROL> <ALTERNATE> <F10> u isto vreme, ostaje u „makintoš" modu i na ST-a se možete vratiti samo isključivanjem računara. Još jedna razlika od *Magic Sac*-a je u tome da umesto da prikaže flešujući karakter u gornjem desnom uglu kada je potrebno izbaciti disk, *Aladin* treperi svetlo za drajv, pri čemu se čuje isprekidani beep zvuk. Softver se može prebacivati samo preko RS232 kabla specijalno prepravljenog za taj posao i ne postoji mogućnost priključivanja *Translatora*. Nismo imali priliku da proverimo kompatibilnost *Aladina* i *Magic Sac*-a.

Ako do sada niste koristili nijedan emulator ili smatrate da vam oni nisu potrebni, samo zamislite naslovni ekran *Aladina* sa porukom „Welcome to Macintosh" na vašem ST-u.

Dalibor Lanik i Dušan Dimitrijević

Moduli za modul

U „Računarima 40" („Kraljica programskih jezika") objašnjavali smo da je snaga modula-2 leži u velikom broju pomoćnih modula kojima se jezik može proširiti na proizvoljan način. Moduli su idealan koncept za tzv. softverska komponente. Kupite procesor reči i ubacite ga u program; kupite bazu podataka, ubacite je u isti program — i imate i editor i bazu podataka. Programiranje korišćenjem softverskih komponenti svodi se na prilagođavanje priručnika i odabiranje najboljeg naredbe (procedure, modula) iz zbirke modula. Ovde opisujemo kako se na moduli-2 uz pomoć komercijalna zbirka modula „ModBase" može rešiti čast i tipičan zadatak unosa robe u magacin.

(3) Kompanija za proizvodnju i prodaju softverskih komponenti brzo se gase i to je tačno idealno bi bilo kupiti i programaka naredbe (source code) modula na samo OBJ datoteka spremna za povezivanja (linkovanje) Skoro da i nema firma koja bi prodavala svoja programa u izvornom obliku

(4) Tuda komponanta su leška za učenja Pa jesu Dobra zbirka modula ekvivalantna je problamaki orijentisanom programskom jazlku

(5) Nija uvek sigurno da će komponante raditi sa datim pravodiocem No to je problam i samih proizvođača prevodilaca Nadavno je Microsoft postavio standard za prenos parametara izmadu programa i potprograma konsti se Pescal konvencija i to u svim programima koji treba da rada pod operativnim sistamima *Windows* i OS/2 Znači ubuduće sa traba bavtli samo prevodiocima koji mogu da koriste Pascal konvenciju

(6) Šte ako programar hoće da dalja proda program sa kupljanim modulima!? Zalsta nepodnošljivo bi bilo plaćati licence za svaki novi program u koma učestvuju ko zna čiji sve moduli

Svi ti prigovori stoja Pogledajmo kako ih je rešila firma PMI Odgovor se svodi na jednu jedinu rač KVALITET Na primar svakoma šalju kompletnu dokumanteciju i damo diskate i to besplatno Cens svih proizvoda kod njih su od 50 do 75% manja nago kod konkurencija Da bi se olakšalo učenja sveka zbirka sadrži modula od vrlo visokog do krajnja nlakog nivoa u odnosu na hardver I — a i o ja veoma ratka pojava — PMI prodaja svaku zbirku i sa izvornim kodom i već spramljanu za povazivanja se naikom od pet najrasprostranjenijih modula na tržištu Prednosti su ogromna Izvorni kod se može čitati kao informacija III se iz njaga mogu izvlačiti važna programaraka tehnike Programar može iznova da praveda module i da ih optimizuja na brzinu izvršavanja III što kraću dužinu programa Ostavljana ja sloboda programaru da manja pravodioce Ispravlja greška ubacuja nova ideja prenosi ceo program na druga mašina druga operativna sistema III čak na druga jezika Štaviše PMI dozvoljava drugim softverskim kućama prodaju svojih modula u izvornom obliku — pod uslovom da obogate originalnu zbirku i naravno plate za to pravo

Sva to zvuči bujno pa pogledajmo kako jedna konkralna softverska komponente izgleda u praksi

Zbirka gotovih modula

ModBase ja zbirka od 17 modula za samóstalno prsanja baza podataka I/III čitanja i pisanja datoteka u formatu dBASE-a U osnovi mehanizma baze podataka leži dobro poznata programaraka tehnika B+Tree pristupa podacima Maksimalan broj adresivih slogova je nešto preko dve miliona lako se *ModBase* može koristiti i samostalno ipak najviše smisla ima kao zamena za spon dBASE U programarskom pogledu *ModBase* je zbirka modula koja se zasniva na drugoj zbirki modula u ovom slučaju *Repertoira* PMI zalsta koristi sopstvena proizvoda što je test kojag nače proći svaka softveraka kuća!

ModBase se naravno isporučuja kompleten i koristi i bez *Repertoira-a*

Ideja vodlija u kraranju naredbi ModBase-a bio je sam dBASE Imana su što je moguća sličnija Na primar naredbe ModBase-a Say(row col s) Gat(row col s) i ReadGats bi poznavaocima dBASE-a trebala da budu poznate bez ikakvih dodatnih objašnjavanja Što je još bolja naredbe *ModBase-a* daju iste maska na ekranu kao i sam *dBASE* skoro bez razlika To znači da biste mogli svoja stara i spora dBASE aplikacija da zamanite novim i bržim a da korisnici ništa i na primate (osim brzina narevnol) Ključno pitanja je koliko brzo ja brzo!? U Računarima 40, str 29 apomanuta je aplikacija koja je na dBASE-u prvo radlia 180 minuta pa je zatim u okviru samog dBASE-a dužina rada smanjana na 90 minuta Onda je program prepisan na modulu-2 i *ModBase* Radio je 28 sekundi za test primer Program je pravedan još jednom se uključenom opcijom za korišćenja matematičkog koprocesora Posle toga test primer sa izvršavso za 17 sekundi Konačno prave obrada trajala je 2 minuta (3—4000 slogova) od čaga ja pola minuta otišlo na štampanja izveštaja Dakle jsate brzo u terminima mašinskih resursa A koliko ja brzo u pogledu utroška programarovog vremana? Cao program (oko 400 linijaj bio je napisen za dve dana i odmah ja proradio

Keko kupiti ModBase

Adresa firma je PMI 4536 S E 50th Portland OR 97206 USA. tel 991 (503) 777--8844, BIX pmi Compuserve 74706, 262 Prmaju direktne uplate čekom a najbolje je poslati broj kreditne kartice (Visa MasterCard American Express) Program stiža na kuću u roku od tri nedelje od slanja broja kreditna kartice

PMI softverske komponante rade u okviru PC standarda Da bi ih koristili treba da imate prevodilac za modulu-2 PMI podržava sve svoje proizvode za sledeći pei verzija modu la Fitted Software Tools Modula-2 Interface Technologies M2SDS JPI Modula-2 Logitech Modula2/86 i Stony Brook Modula-2 Neki proizvodi mogu sa dobiti i u verzijama Farbware Modula-2 FTL Modula-2 PCollier Modula-2. Obj datoteka se međutim mogu koristti sa svakim pravodiocem (ne mora uopšte biti modula-2 u pitanju!) koji je usaglašen sa operativnim sistemima Windows i/iii OS/2

Uz svaki program dobija se uputstvo kao knjiga a takođe i kao datoteka na disku

Cena softverskih komponenti su Repertoira $89 ModBase $89 EmaStorage $49 Repertoira/Btrieve Toolkit $149 Graphix $199 DynaMatrix $69 Macro2 $89 Make/Xref $89 Na navedena cena za izvorni oblik modula treba dodati $10 za poštarinu po paketu

Vaiike softverske kuće bi htela da ubeda sav svet da sa nove aplikacije moraju pisati na C-u da ih moraju stvarati timovi od desetina i stotina programara da moraju biti dugačka ber 100000 programskih linija (po mogućstvu i višaj ta da — zbag svega toga — moraju biti mnogo skupše. Dva najvažnija primara su OS/2 firma Microsoft (razvoj dug skoro dve godina preko 1500000 programskih linija u C-u) i *dBASE IV* firme Ashton-Tate iz koje je nakoliko glavnih programara jednostavno — pobeglo U budućnosti stvari će biti drukčije Microsoft je nedavno objavio da će u sve svoja jezika (pa čak i u bejzik) ubacivati alamanta objektnoorijentisanih jazika iako je to najbolji put ka produktivnosti! (osobina naslađivanja i polimorfizma vidi Računare 38, str 25-–27 i Računara 36 str 25—30j na to ćemo najverovatnija pričekal! U međuvramanu pitanja programerska produktivnosti ostaja i dalja otvorano

Softverske komponente

Jedno tradicionalno rešenja nalazimo u operativnom sistemu UNIX Pošlo u njemu istovremano može da radi i više korisnika I više programa postavilo se pitanja kako da programi maduaobno sarađuju U tu svrhu izmišljani su cevovodi (pipes) i filteri Ideja je da se piše niz po obimu ili po koncepciji malih programa koji bi izlazna rezultate ostavljali tamo gde neki drugi program može da ih pročita. Na primar program smašta rezultete u datoteku na disku. a sledeći program u „cevovodu prosto kopira datoteku na štampač Takvi programi se nazivaju alatima (tools) — po analogiji sa opipljivim alatima u gradavinarstvu Atali i cevovodi zalste povećavaju produktivnost ali odakle nam junika na kućnim i PC računarima!?

Treći način za podizanja programarske produktivnosti su softverska komponante Poznate au još pod imanom programskih biblioteka (toolkit) Prodaju se kao dodaci za bukvalno svaki jezik na tržištu počev od bejzika do ada Ta ideja je toliko močna da je prvo C a po ugledu na njaga i modula-2 bio zsanoven upravo na toj koncepciji U C-u i modull-2 se čak ništa na bi moglo ni odštampati bez uvoza spoljnih funkcija iil modula! (Vidi Računara 40 str 27—28) All mana traba pretvarati u prednosti pa sa čak može smatrati da je jedan od glavnih razloga popularnosti obaju jezika upravo to što nazavisnim kompanijama ostavljaju prostor za prodaju dodatnih lamatskih biblioteka funkcija i modula Broj dodatnih biblioteka važan ja i pri opradatjvanju za neki jezik umirujuće daluja saznanja da ćate moći da kupite naredba za sve počev od editora preko baza podataka do programa za telekomunikacija i igranja bridža III goal

Pa ipak mnogi programari namaju povaranja i iskustve se softverskim komponantama Dva su osnovna problama (1) kako uskladiti sopstvani atil pisanja programa sa ludim naradbama i (2) da il su već napisana zbirke koja rašavaju naš naposredni problam Odgovor na obe pitanja zavlal od koga nabavljata softverska module Ovda ćamo prikazati zbirku modula za čitanja i ▶

▶ pisanje datoteke u dBASE formatu a autori su saradnici sva poznatija američka softverska kuća PMI

PMI i MI

PMI prodaje samo softverske komponente za modulu-2 Posluji preko dve godine Prvi proizvod bio je *Repertoire* izuzetna zbirka modula od preko 800K izvornog programa za baze podataka definisanje ekrana liste i slično Danas imaju desetak zbirki modula *(DynaMatrix Btrieve ModBase Macro2 Make/Ref EmsStorage Graphix* i još neke) Osobenost firma PMI je što jako dobro shvata da su njihovi kupci programeri i to varovatno sasvim kompetentni u svojoj oblasti A programeri su znamo nedobudni i gaje predrasude prema tudim programima Evo nekoliko tipičnih sudova (a u nekim slučajevima i tačnih konstatacija) o kojima treba povesti računa pre korišćenja kupovne zbirke naredbi

(1) Lakše je i sigurnije pisati sopstvena modula nego zavisiti od tuđih To je svakako tačno ell vremena se menjaju Današnja aplikacije moraju biti jednostavnije za korisnika, brže u izvršavanju zasnovane na grafici i — povrh svega — treba ih napisati za što kraće vreme Na primer jedan programer na može više ni da nauči ave o grafičkim adapterima za PC računara a kamoli da sav softver piše iznova Jedino zato datoteka zatvori naredbom CloseDBF i odmah zatim otvori sa OpenDBF Deluje kontraproduktivno ell radi u svim slučajevima! Zatim sledi otvaranje indekana datoteka naredbom Openin-

Magacinsko poslovanje

Umesto suvoparnog nabrajanja naredbi *ModBase-a* ovo kompletno rešenog primera sa programom Problam je samo promanili opis slogs I iznova prevsali program na moduli-2 Korišćena je *Logitech Modula-2 ver 3 03* (za detaljen opis vidi Računare 39 str 19—21)

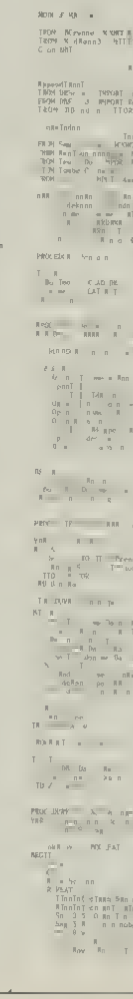
U računar sa unose podaci o vrsti robe (numerički šifrirano) i količini pri svakom ulazu u magacin U svakom trenutku potrebno je znati količinu unate robe Zaalak sa rašave formiranjam sloge od dve polje Prvo polje (Vrsta) ime desal maste s numeričkog se tipa (celobrojnog) Drugo polja (Količina) numeričkog se tipa I ima dsasl mesta od čega su dva za cifra deano od decimalnog zaraza indakalranja sa vrši listovrameno es unosom podataka Prairažlvenja i sabiranja količina robe obevlja se leko šlo se u lancu indaksa pronađe prvi indeks za koji postoji tražana vrsta robe Ukoliko takva robe nama javlja se odgovarajuća poruke Ako robe ima pronadana vrednosl u polju Kolicina dodaje se na promenljivu KoliVrsl Završetak sabiranja količina robe je onog trenulke kada sledeći indeksni slog nama vrednosl Vrsle jesneku sa traženom vrednočču Na kraju ee dall zbir Ispisuje na akranu

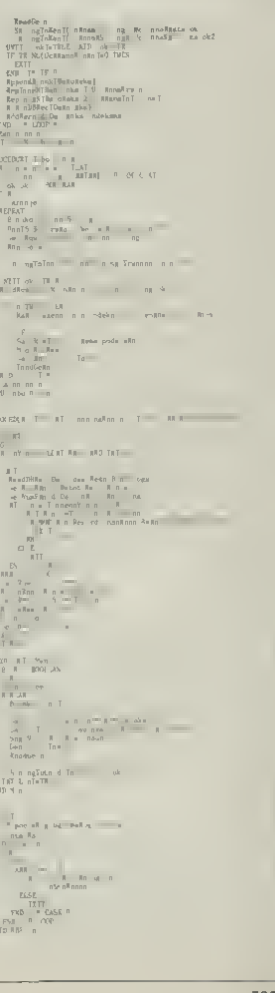
Uvoz procedura Obevlja se iz pat *ModBase* modula (DBCreate ModBase3 DBScreen DBFiala DBIndax) jednog modula iz Repertoira-a (SmartScrean) i čelirl modula iz Logltach Module-2 ver 3 03 (RealConvarlons Real InOut NumberConversions SirEdli) Ovl moduli se eksplicitno uvoza a pri povazivanju (linkovanju) moža se uočiti implicitno pozivanja čitavog niza drugih modula iz Repertoire-a i sama Moduia uključujući i modula najnižeg nivoa koji direktno opšta sa hardvarom

Globalne promanljive Promanljive Datoteka je tipa DBFile (Tipovi podeleka dell eu u zasebnoj tabell) Promenljive indekana ja tipa DBIndex Promanljiva Dosime je korlaničko ime za promanljivu Datoteka preko koga se komunicira sa hardvarom Promanljiva Inima je korisničko ima za promanljivu Indekana

Opis procedura U proceduri Postavka postavijaju se Dosime i Inime Druge mogućnosl je da korisnik sam učlava željene imene U proceduri DefinicijaSloga ugnježđana je procedura DefinicijaPolja kojom sa dalnišu sva karakteristika promanljive Opis tipa DBFialdDescriptor Sama procedura formira datoteku na disku uz pomoć naredbe BuildDBF Procedura Blanko postavlja slovni niz proizvoljna dužina Tu se vidi jedna od odličnih osobina Modula-2 — u proceduru se može preneti bilo koje dužine nize jer posiopl funkcija HIGH koja zna tačnu dužinu niza iz tekala programa vidi se da se procedura Blanko poziva sa više mesta i sa različitim dužinama argumenata

Procedura Otvaranja ime zadatak de otvori I DBF I indeksnu NDX datoteku Postupak ja sličan i počinja pokušajem otvaranja DBF datoteka pomoću naredba OpenDBF Ako datoteka se imanom Dosima postoji na disku onda je postupak završen Ako ia datoteka nama onda logička promanljiva opan iz sloga DBFila ima slaius FALSE pa se poziva procedrue DefinicijaSloge koja traba da izgradi prvi slog u DBF datoteci Prema priručnicima to bi trebalo da buda sasvim dovoljno Ali nije moguć upis podataka pa se





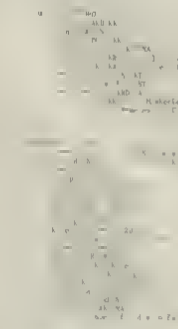
rešenje je kupovina gotovih modula — kondenzovanog ludog znanja

(2) Komercijalna zbirka modula su preskupa To je na žalost tačno ali sa često previše Na primer, *Turbo Pascal* sa reklamira kao jeftin program, ali cena svih dopunskih zbirki softverskih komponenti popela bi se na 5—600 dolara Nijedan program koji računa na širu popularnost ne bi smeo da košta više od 100 dolara niti bi proizvođač komponenti smeo da ima kopirajli na finalni proizvod

sex koje se izgraduje naredbom BuildIndex ukoliko već ne postoji I to za redni broj polja u slogu koji je odražen trećim paramterom ove naredbe U slučaju da DBF datoteka nije otvorena odgovarajuće indeksna datoteka ne može se izgraditi Za izgradnju indeksna datoteke dovoljno je da je DBF datoteka otvorena, pa makar bila i bez podataka Ovo je također vrlo lepa osobina

Procedura Zatvaranje istovremeno zatvara I DBF I indeksnu datoteku

Procedura UcitNovihSlogova poziva proceduru Otvaranje u beskonačnoj LOOP-petlji učitava podatke I poziva proceduru Zatvaranje.Učitavaju se dve podatka — UcitaneVrata I UcitaneKolicina S obzirom na osobinu naredbe Get — da može da učitava isključivo slovni niz (string) potrebno je konvertovati podatke Konvarzije se vrši naredbom StringToReal iz module RealConversions gde je prvi parametar — učitani slovni niz drugi — učitani podatak kao REAL I traći — logička promenljiva Proces sa ponovlje sve dok obe logičke promenljive nemaju vrednost TRUE odnosno sve dotle slok se ne učitaju resini brojevi Naredbe Say I Get koje se primenjuju za ovo učitavanje u potpunosti oponašaju odgovarajuće dBASE naredbe a naredba ReadGets oponaša dBASE naredbu READ Naredba ReadGets omogućava izmenu oba podatka pra nago što se pritisne Enter Izlazak iz beskonačne LOOP-petlje odvija se kada je UcitaneVrata jednake null Radi sigurnosti u poredenju brojeva naredbom Trunc promenljiva UcitaneVrata tipa REAL pravodi se u tip CARDINAL Ako je izabrana bilo koja druga vrata DBF datoteci se dodaja prazan slog naredbom AppendBlank Naredbom ReplaceN pridodeljuju se promenljive UcitaneVrata prvom polju a UcitaneKolicine — drugom polju Redni broj polja koje se pridodeljuje definisan je drugim argumentom te naredbe Slog se na disk upisuje isključivo naredbom WritaDBRec Naredbom AddRecord ažurira se stanja u indeksnoj datoteci



Procedura IzborVrste ima zadatak da nad otvorenim DBF I NDX datotekama e za učitanu vrstu naredbom FindPositionN pozicionira indeksa na prvu pojavu traženog sloga Ako takav slog ne postoji šalje se poruka o eko postoji pozive sa procedura Rad U tom smislu interesantno je da drugi parametar u pozivu procedure Rad (a on glasi indeksne curreniKey racorcinum) sadrži podatak o aktivnom ključu i trenutnom slogu u indeksnoj datoteci

Procedura Rad inicijalizuje promenljivu KolIVrat na nulu Zaobilazno postavljanje rasinih brojeva preko funkcije FLDAT i VAL je specifičnost *Logitech Module-2* i nema nikakve veze sa rešavanim problemom U *JPI Moduli-2* mogio bi sa pisati i klasično KolIVrsi = 0 0

U beskonečnoj LOOP-petlji naredbom ReadDBRec čita se slog Podatak posteje dostupan programu tek posle naredde GetNumField kojom sa apscitiCira DBF datoteka radni broj polja I promenljive u koju sa podatak unosi Sve dok je promenljive Vrata jednake zadatoj promanljivoj TrazaneVrata tekuća količine dodaje sa na ukupnu količinu U suprotnom napušta sa petlje Vrlo je interesantna logičke funkcijska procedura NaxtRecord koja pomere indeka na sledeći slog Ako sloga nema izlazi se iz beskonačne LOOP-petlje Na taj način spoljašnji IF uslov kontrolise da li je došlo do promena vrste a unutrašnji IF uslov — da li se došlo do kraja datoteke Po izlasku iz petlje ispisuju sa vrsta i količina rode Naredde WriteReal direktno ispisuje rasine brojeve dok bi za primenu Say naredbe bile potrebna konverzija iz resinih brojeve u stringove

Procedura Meni ispisuje jednostaven meni i učitave korisnikov izbor S obzirom na prirodu CASE naredde potrebno je da sa slovni niz pretvori u CARDINAL naredbom StringToCard Na zaboravimo da je u moduli-2 obavezne upotrebe rezervisane rači ELSE na kraju naredde CASE

Ovaj program je lako proširiti na veći broj polje u slogu indeksirati po većem broju ključeva i usložnili sam metamelički model traženog izlaznog razultata npr traženjem iroškova (= cena • količina) prosečnih cene ravalorizacija i drugih tipičnih zadataka vezanih za baze podataka odnosno magecinsko poslovanje U odnosu na *dBASE* nedostatak *ModBase*-a je da ne može neposredno da indekaire po slovnom zbiru dveju ili više polje Konkratno ako su podaci o imenu i prezimenu datinisani u dve odvojena polje *ModBase* može direkino da indekaira aveko od ih polja ponaosob Ako je potrebno indeksirati po slovnom zbiru imena i prezimena, tada treba otvoriti dodatnu DBF datoteku u kojoj bi se nalazili podaci u slovnom zbiru imena i prezimena Te DBF datoteke sa tada može indekalreli po tom jednom polju a dobijena NDX datoteka može sa primeniti i na originalnu DBF datoteku u kojoj su ova polja bila posebna

ModBase je prvenatvano namenjen svim onim posednicima masovnih DBF datoteke koji ne mogu biti zadovoljni brainom izvršavanja programa u dBASE-u Po samim nazivima naredbi jasno se vidi da je *ModBase* namenjen korisnicima dBASE-a koji bi želeli da iskoriste svoje datoteke i sa minimumom napora pradu na kvalitetniji programski jezik uz brže izvršavanje avih aplikacije Naravno kao što se i iz ovog programa može videti *ModBase* može biti osnova za samostalni razvoj programe čak i eko namete datoteke u dBASE formatu Resino vreme razvoja ovog programa je jeden čovek-dan Ualoženjavanja problema npr povećavanjem broja polje I/III proračune zehvaljujući izvanrednom Poini editoru (sastavni deo *Logitech Module-I* na povećave bitno vreme razvoje programe Ako nemate dBASE datoteke bolje je da sa opredelite za *Repertoire* o čijim prednostima će biti više rači nekom drugom prilikom

Duško Savić

Šta ima novo

Hardver

ST SCAN

ST Scan je novi proizvod firme *Navarone Industries* i omogućava skaniranje teksta i integralije na eteriju ST Ima rezolucije od 75 100 200 ili 300 tečaka po inču Brzina skaniranja je 12 sekundi po slanu pri maksimalnoj razolucoji Skaner radi sa 32 nijanse sive boje ili u crnobelom Priključuje se na ST-ov kertridž port i zahteva (minimalno) 520 ST sa drajvom i bele kojim monitorom Skanirane slike i tekst sa mogu učitati u *FSP Publishing Partner Easy Draw* ili *DEGAS* Cena ovog uredaja je 1239 USD (Navarene Industries Inc 454 Kenneth Ave. Campbell CA 95008)

Komercijalni softver

EASEL ST

Ako vam je desadlie sive boje pozadine ST-ovog desktop-a kako bi bilo da umesto toga imate bilo koju sliku o *DEGAS* ili *Neo* formatu *Easel ST* vam omogućave da prilikom buderia računara zamenite sivu pozadinu desktopa slikama iz *DEGAS-a* ili *Neochrom-a* *Easel ST* radi u sve tri rezolucije pri čemu se mogu delimisan različite slike za sveku razoluciju Novi desktop se pojavljuje sa svim uobičajenim ikonama prozorima i odabranom slikom koo pozadinom Umesto tradicionalne fotografije parodice na stolu možete staviti digitalizovanu verziju na desktop (*Easel ST* Computer Fensatrations P O Box 151 Lake Menoroe FL 32747)

Periferijska oprema

Modemi

Na tržištu sa nedavno pojavilo nekoliko veoma dobrih modema za ST Prvi je *Avatex* 1200E — omogućava brzinu od 300/1200 bauda automatsko odgovaranje poboljšanu komunikaciju ĆCITT operacija i 7 prakideče za podešavan e Cene je veome prihvatljiva — samo 99 dolara (408-732 1181 USA)

Drugi je *Supra modem* 2400 sa asinhronim pranosem od 300/1200/2400 bauda sa automatskim odgovaranjem u tonu III impulsu dva modularna telefonska konaktora i podešavajućom jačinom Podešavanje se vrši pomoću softvera i smešta se u nezbrisivu memoriju Cena je 180 dolara (503-967-9075 USA) Koliko para toliko i muzike!

Hardver

Miš menadžer ST

Ova vest obradovaće sve starISTe koji imaju ugraden (interni) flopi Poznato je da se kod ovih modela konaktori za miš i džojsiik nalaza ispod kompjutera sa donje strana pa vam sveko priključivanje pokvari raspoloženje Firma *Practical Solutions* je proizvela *Mouse Master* koji se jednostavno priključuje sa dve standardne stari konaktora na kompjuter a omogućave priključenje dva džojstika i miša Jednostavnim preklopnikom regulišamo da li će na portu i biti priključen miš ili džojsiik Cena je 34 US dolara. Možeia ga naručiti na telefon 602-884-9612 USA.

Softverski podsetnik

Spektrum

3D GAME MAKER

Tržišta igara preplavljeno je programima sa trodimenzionalnom grafikom. Koliko puta ste igrali neku stotu verziju „Knightlore -a, i proklinjali programera zato što na može da smisli ništa novo? E pa ako na može programer možete vi i to uz pomoć programa Kristofera Hajverda (Christopher Hayward) — „3D Game Maker. Program ima tri dela.

Prvi deo — 3D GRAFIX EDITOR — Ovaj deo služi za izradu sprajtova i grafika koju ćete koristiti u igri. Ekran je podeljen na dva dela u levom se nalazi uvećlćana slika predmete ili lika koji trenutno editujeta. Sa desna strana su razna korisna naredbe:

NEXT i BACK — njima birate sličicu koju želite da editujete

MASK — na gornjim slikama u prirodnoj veličini iscrtavate masku novog sprajta

BASE — uz pomoć ove naredbe svaku tačku na uvećlćanom crtežu videćete prikazanu u 3D tehnici

FLIP — rotira sprajt za 180 stepani

ERASE — briše trenutno nacrtanu sliku

FILL — popunjava deo slika bojom

ABORT — poništeva dajstvo svih prethodnih naredbi

EXIT — povratak u glavni meni gde možete snimiti ili učitati nove sprajtove

Naravno možete podseti boju slika kao i pomanti sprajt na ava strana pomoću slralica

Editovenje se vrši tako što kurser namestite na bilo koju tačku uvećlćana slike i pritisnate pucanja. Tačka će promeniti boju a vi možete da nastavite sa drugim delom

Drugi deo — 3D ROGM EDITOR — U ovom programu pravite mapu igre i u sobe postavljate smeteia. U sredini je soba koju obradujete, gore je pokazivač memorija a u donjem delu ekrana su (s leva nadseno)

— Mape sobe u kojoj se nalazite

— Visinometar — pokazuje na kojoj visini je vaš sprajt

— Mapa igra (16×16 polja) podeljena u 4 dela pokazuje u kom se delu nalazite

— Mapa igre (8×8 poljaj pokazuje sve sobe kao i predmete u njima

Kursorom pomarate predmet koji ste odabrali. Na zamiju ga spustite se ENTER a podižete sa SPACE Tasterom M birate jedan od 20 ponudanih predmata. Tuklanje sva pomoćna sredstva se ekrana a „X vza vraća u početni meni, iz kog opet sve štu ste da sada uradili možete spramiti na kasetu. Treba znati i da za penjanje treba pritlanuti pucanja+dola a za spuštanja pucanja+gora

U sobu na mapate stipati vita od dvadesat predmeta, a nova vrata ćete postaviti tako što dodate u sredinu ivice sobe (posložite sa mapom) i prilianete pucanja. Keda i tu sve sredite učitajte poslednji deo

Treći deo — 3D GAME MAKER — Na samom početku imata mogućnost da učitata nova sprajtove i nove sobe koja ste edilovali u pretnodnim programima. Naravno možete i odigrati igru. Kraćete sa po poznatoj šemi da se prvo okrenete na željenu stranu i onda prilisnete napred. Možeta i skakati (pucanja) ili pucati (nazad). Ako je sve u radu pritisnite taster S, napišite ima igra i autora i animite igru.

Vaša igra napravljena „3D Game Maker -om naće imati naka ajana originalna rešenja jer sa sve svodi na jurnjavu lavirintom i izbegavanje napriatelja ali će ipak biti uradena na visokom nivou jer program ima zanimljive zvučna alaka i vrlo dobru animaciju.

Bojan Majer

Atari ST
Flight Simulator II

Program Flight Simulator II u verziji za Atari ST pojavio se na tržištu nagda pred kraj 1986 godina i dosad je još uvek napravezidni simulator leta za kućna računara. Nadugo potom je izašla i verzija za amigu, koja je praktično identična se verzijom za Atari, tako da ovaj članak može da buda od koristi i vizanicima „amiga.

Verzija za dva pomanuta računara predstevijaju značajan napradak u odnosu na verzija za IBM PC. C64 III Atari 800 koje je late firma — SUBLOGIC. Izdala pod istim imanom.

Izdato je nakoliko verzija ovog programa. Prvo su se pojavile posebna verzija za kolor i monohromatska, moćnore a kasnija i verzija 1.1 koja na jednoj jednostrano) diskati sedrži obe verzije i još naka manja poboljšanja.

Nakon učitavanja programa na akranu se pojavljuja pogled iz pilotskog sedišta se instrumentima i pogledom napred a sesvim gora su propadajući manji. (ovde za razliku od GEM-a nija dovoljno kursorom samo dotaći opciju već treba i klknuti levo dugmaj. Ukoliko se miš na pomara u roku od 30 sekundi počinja linl damo, koji možete prakinuti prilskom na levo dugma miša i klkom na mali pravougaonik u levom gornjem uglu (Close box).

Program simulira dva tipe aviona jednomotorni propelaraš Cesana 182 i mlazni avion (business jet) „Gates Learjal 25 g"

Kod propelaraše je naglasak na šte vernijoj simulaciji dok ja mlaznjak više namanjen za zabavu nego za verno simuliranja. Početnicima prediažem svakako propelaraš koji se dalako lakša upravlja pogotovu pri slatanju.

Postoji i opcija nazvana „As prvog svetskog rata (World War I Aca) gde ulaćete u vazdušnu borbu ali ona nija tako dobro obrađena. Postoja programi za borbenu simulaciju leta koji su bolji za ljubitelje takvog tipa zabave.

Moguće je upravljanja tastaturom, mišem i kod verzija 1.1 i džojstikom. Upravljanja mišem je najpracizniia pošto se njima prilično dobro simulira analogni džojatik koji je najprikladniji za ovakve programa te ga predlažemo svakoma.

Motor već radi ia treba samo dodati gas kliknite desno dugma (na mišu) čima ste iskliučili kursor i prebacili se u upravljački moć zatim pritisnite levo dugma i za prilisnutim dugmatom gurajta miš gora (od sebe) gas i broj obrtaja će se povećavati, avion će počati da rula po pisti. Dodajte pun gas otpustite dugma i seda miš simulira upravljačku palicu u avionu znači levo—desno za pravec a gore—dole za penjanja/spuštanja, kao kod svakog aviona miš treba vući ka sebi za penjanja. Nzalojte blagim pomeranjem miša levo-deano održali avion na sredini plate. I kad dostigna brzinu od oko 60 čvorova (propelaraš) povucite miša ka sebi oko 3 cm. Avion će polako početi da se penja. Ovo je tačka gde mnogi graše poviaćeći odmah palicu prama sebi dok avion još nema dovoljnu brzinu za ošro penjanja što se završava stallingom (propedanjem) i uništenjem aviona.

Sad možete da razgledate okolinu prilskom na tastera za izbor pravca gledanja odnoano izborom iz manija „VIEW. Ukolko nastevila da lalila pravolinijski za par minuta ćete ugledati grad San Francisko se naizbežnim „Golden Gate -em. Ovaj grad je inače najdetaljnije prikazan u programu.

Istovramano magu biti otvorana tri prozora primerom na tastere F1 -F3 (lalim tastarima se prozori i zatvarnjuj. Sa F3 se otvara prozor za mapu predsta iznad koga latite. Svaki prozor se može zumirati u velikom opsegu (pogledaj spisak komandi).

Prozori se mogu povećavati i smanjivati na način kao kod GEM-a ali da bista imali pogled napolja prako celog akrana prvo morate ukloniti tablu as instrumantima. Pomarite kursor ka sradini gornjeg dala table i „povucite ja dole. Tablu možete vratiti klikom na donju ivicu akrana.

Svi znaju da ja slatanje najteži deo pilotiranja, pe ćemo dati nakoliko saveta. Zbog vrlo dabrog 3D prikaza sasvim tino se može slatati i bez ILS-a (Inatrumant landing system). program dozvoljava da slatila i pared plate.

Najvažnlja je pri slatanju tačno kontrolisati brzinu i horizontalnu, i vertikalnu. Slatanje započnite na dovoljnoj udaljenosti od aerodroma kako biste imali vramana da se

Pregled komandi

Pogled
F1 uključi/isključi glavni 3D prozor
F2-uključ-/isključi drugi 3D prozor
F3-pokaži/isključi mapu
F8-zum povećanje
F10-zum smanjenje
— tini zoom povećanje
( ) — tini zoom smanjenje
Esskspace-reset zoom
T-pogled napred
B-pogled nazad
F-pogled levo
H-pogled desno
G-pogled dole
R-pogled napred—leva
Y(Z) — pogled napred—desna
V-pogled nazad—levo
N-pogled nazad—desno
S-pogled iz prateće aviona
D-pogled iz prateće vozila
C-pogled iz lomja
X-pogled iz kabine
Kurzorski tasteri — fino podešavanja ugla gledanja
Clr Heme-resetovanje ugla gledanja

Miš
Gore — nse dela
Dole — nas gore
Levo — zaokrai ulevo
Desno — zaokrai udesna
Levo dugme i gora — dodavanja gasa
Levo dugme i dole — smanjivanja gasa
Levo dugma i levo — kočenja (samo ne zemlji)
Levo dugma i desno — otupuštanja kočnica
Deano dugme — biranje kurzor moda III moda upravljanja

Specijalna komanda
Help — objašnjenja pojedinih instrumanata i komandi
U — podizanje/spuštanje stajnog trapa
K(0) — zakrilca gora
J(*+) — zakrilca dole
<( ) — kormilo levo
>(-) — kormilo desno
P — pauza
O — senčana/linijska grafika
L — uključivanja svetla u kabini
Q — sečuvaj trenutnu silisciju
A — vrati snimljenu situaciju
Z(Y) — autopilot da/ne
Tab — zvuk/bez zvuka
J — upravljanje džojalikom (kao kod verzije 1.1)

Upravljanje
Numerička tastatura
9 — dadaj gas
3 — smanji gas
1 — kočenje
2 — nos gora
8 — nos dole
4 — zaokrai levo (eleroni)
6 — zaokrai desno (eleroni)
0 — kormilo levo
— kormilo desno
5 — cantriranje kormila i elerona

Samo u relnom modu (WVI Ace)
Shift i W — objave rata
Shift i E — rainl izveštaj
Shift i X — bombardovanja
Razmaknica — pucanja

Komandni tasteri su dati za englesku tastaturu. Pošto kod nas većina ima nemačka u zgradama je dsl i odgovarajući taster za nemačku tastatura tamo gde ima odstupanja.

postavite u pravac piste obavezno izvucite flapsove bar do drugog podeoka, naravno izvucite i stajni trap (točkove) polako se spuštajte i smanjite gas tako da vam brzina bude oko 65 čvorova Nosem ciljajte početak piste za sletanje i regulacijom gasa držite brzinu Pred sam dodir piste podignite nos malo gore radi što mekšeg sletanja Izbegavajte grube manevre jer avion ne reaguje momentalno na sve komande Malo kome je uspelo da ispravno sleti u prvih par pokušaja pa se nemojte brzo razočarati

Ovde treba napomenuti da visinomer pokazuje nadmorsku visinu (u stopama) a ne visinu od tla te je poželjno znati nadmorsku visinu aerodroma na koji slećete

Ovo je jedan od retkih programa gde je taster HELP vrlo dobro iskorišten Prilikom na to dugme kursor se pretvara u znak pitanja i pomeranjem i klikom na odgovarajući instrument odnosno tačku menija biće ispisani svi važni podaci

U opciji ENVIRO menija možete podesiti vremenske uslove i godišnje doba detaljno objašnjenje nije potrebno pošto se sve vidi iz samog menija

Sledeća tačka u glavnom meniju je SIM tj kontrola simulacije Najinteresantniji su Reliability (pouzdanost) i ,Realism što je pouzdanost manja češće će otkazivati pojedini sklopovi aviona najčešće instrumenti Sa opcijom Realism podešavate neke specijalne parametre kao na primer da li da postoji detekcija pada (sudara), način paljenja motora itd

Područja po kojem možete leteti sa ovim programom je ogromno, praktično obuhvata celu SAD dok su detaljno obrađena pet područja San Francisko i okolina (oko 200·200 milja) Los Anđeles i okolina (80·100 milja) područje oko Nju Jorka i Bostona (200·150 milja) područja oko Čikaga (80·150 milja) i područja oko Sietla (100·100 milja). Aerodroma na koje možete sleteti ima ukupno 120 U prva četiri područja možete direktno ući iz menija Select prerecorded , a u Sietl sa Position set koordinate su 21343-6584 Naravno možete i preleteti recimo iz Nju Jorka u San Francisko preko Čikaga uz malu pomoć karte SAD Uz program (original) se dobija i šest mapa navedenih područja sa označenim aerodromima i radio farovima sa njihovim frekvencijama, što je neophodno za iole ozbiljniju navigaciju

U novije vreme pojavili su se i razni , Scenery disk -ovi, na kojima su razni delovi SAD, Japana i Zapadne Evrope

Sistem navigacije uz pomoć radio farova verno simulira navigaciju u stvarnoj avijaciji ali pošto većina kod nas ne poseduje mape nema svrhe detaljno objašnjavati ovaj dosta kompleksan deo programa

Ostaje još da se objasni tačke u glavnom meniju Situation Sa , Select prerecorded možemo da se brzo prebacimo na nekoliko unapred pripremljenih područja sa „Save and name sačuvamo trenutnu poziciju i ostale parametre u memoriji a sa „Recall se vraćamo u sačuvanu situaciju Te situacije se mogu i snimiti na disketu sa opcijom Save RAM to disk Tu je i instant replay sa kojim možemo da repriziramo zadnju minutu leta

U programu postoji još čitav niz opcija od kojih vredi napomenuti sledeće

Pogled iz pratećeg aviona Spot plane Moguće je podesiti mnoštvo parametara kao udaljenost brzinu i pravac praćenja itd

Podešavanje dnevnog vremena vrši se klikom na cifre časovnika „TIME na instrument tabli čime menjate period dana i vidljivost koja je u sumrak smanjena a noću se vide samo osvetljeni objekti

Multy Player omogućava povezivanje dva računara preko RS 232 konektora i let u tandemu

Prvi svetski rat cilj je uništiti neprijateljske objekte sa druge strane reke bombardovanjem i oboriti što više aviona Za poštenu borbu morate prvo objaviti rat pritiskom na ,Shift i W

Autopilot omogućava let na duže relacije bez zamora a sa Slew možete brzo menjati poziciju bez ,stvarnog leta

Instrumenti su pregledni i jasna im je namena problem mogu biti samo dva instrumenta u sredini sa vertikalnim crtama to su dva radiometra (VOR) ali sa njima ionako nećete ništa postići bez mapa sa naznačenim frekvencijama radio farova

Petar Putnik

Amstrad CPC

Music System

Imate CPC 464? Ako je odgovor potvrdan onda i sami znate koliko je amstrad jak na polju zvuka Čak se i iz bejzika može puno toga uraditi Međutim bejzik ma koliko bio dobar ne daje vam onu lakoću i komfor rada kao programi za pisanje muzike Nameće se činjenica da je za ozbiljnije pisanje muzike neophodno koristiti neki program Najbolji koji je meni dopao u ruke a pretpostavljam i jedan od najboljih za amstrad je Music System

„Music System' ima sljedeći izgled ekrana najveći dio zauzima notni sistem (NS) ispod je klavijature (K) a lijevo su (odozgo ka dole) pokazivač količine napisanih nota 3 manja prozora od kojih prvi pokazuje zvučnik (koji se invertuje prilikom reprodukovanja napisane muzike) i broj kanala na kojem se trenutno može pisati. Drugi prozor pokazuje koliko još nota može da stane i takt u kojem se nalazi pokazivač u NS Promjenu ovog prozora vršite sa F Do njega je treći prozor sa metronomom (M) koji naravno, pokazuje tempo Ispod ova tri prozora je ponovo prozor u kojem se nalaze podaci o jačini anvelopi i oktavi koja je važeća za svaki kanal posebno Sada nešto o tipkama V omogućava pristup notnom sistemu M je za metronom K za klavijaturu a C' je za ovaj maločas opisani prozor (možemo ga zvati statusni-ST) Ova četiri mode (NS M K ST) se mogu mijenjati i prilikom na SPACE BAR Primijetićete da je mod koji je trenutno aktivan uokviren u crno

Na dodatnoj numeričkoj tastaturi tipke imaju sljedeću funkciju 1 2 i 3 mijenjaju kanal na kojem se radi Pritisak na 4 vam daje prozor Files koji služi za Load/Save operacije ,5 ispisuje prozor Values a 6' — Commands 7 daje Info prozor u kojem ćete naći informacije o broju napisanih nota, taktova i sl ,8 i 9 služe za smanjivanje/povećavanje određenih vrijednosti npr pritisnite M za metronom pa pritiskajući 8 i 9 pogledajte šta se dešava Ø reprodukuje kompoziciju od mjesta gdje se trenutno nalazi pokazivač pa do kraja Kurzorski tasteri su takođe u upotrebi Kosnite se za ono se šta su i predviđeni pomjeranje note gore/dole pomjeranja po notnom sistemu lijevo/desno i za pomjeranja po raznim prozorima COPY taster ima različitu funkciju u raznim modovima Tako u K modu vrši snimanje onoga što svirate preko tastature

Sada nešto o svakom modu posebno

Notni sistem je glavni mod U njemu pišete note Pisanje je vrlo jednostavno Stavite notu na svoje mjesto pomoću Onda joj odredite trajanje sa „8 i „6 i pritisnite ,ENTER Ako hoćete pauzu pritisnite tipku sa zarezom Sa ,8 i 9 ponovo određujete trajanje Ponovni pritisak na zarez vraća note

Nota pauze povisilice snizilice i sve ostale stvari se pišu u skladu s muzičkom sintaksom Da se podsjetimo osnovna nota je četvrtinka (crna sa stabljikom) Veće od četvrtinke su polovinka (neispunjena bijela sa stabljikom) i cijela nota (neispunjena bez stabljike) Manje su osminka (crna sa stabljikom i jednom zastavicom na stabljici) šesnaestinka (2 zastavice) i 32-ginka (3 zastavice) Pauze imaju istu podjelu Svakoj noti/pauzi možete produžiti trajanje za pola njene vrijednosti tako što ćete joj dodati tačku Dodavanje/brisanje tačke se vrši tasterima na kojima je, naravno tačka Toliko o osnovnoj sintaksi

Ako pritisnete taster COPY dok ste u NS modu dobićete prozor u kojem se nalaze sljedeće opcije **Accidentals** (povisilica snizilica i sl) **Barlines** (linija za završetak takta) **Dynamics** (jačina sviranja) **Notes** (trajanje nota možete postavljati sa ,8 i ,6 ili ovom opcijom) **Envelopes** (postavljanje anvelope samo za jednu notu) **Ties** (postavljanje veza između 2 nota radi produženja njihovog trajanja) **Synthesizer** (mijenjanje anvelopa tona i jačina sa COPY vršite kopiranje ili zamjenu anvelopa i setova) ext (premještanje brisanje i provjera proizvoljnih blokova kompozicije) i **Locate** (traženje anvelopa pauza veza i sl u kompoziciji) Sve ove opcije možete dobiti i direktno pritiskom na A, B D, N E, T S X ili L, respektivno Kroz kompoziciju se krećete lijevo/desno s kurzorima U kombinaciji sa ,CTRL ovi tasteri vas vode na početak/kraj kompozicije Taster DEL briše note a lunta postavlja stabljiku nagore ili nadole „Z i ? odsviraju notu na kojoj je pokazivač. U K modu možete da svirate i direktno unosite note u NS (prvo pritisnite „COPY) Za sviranje se koriste gornja dva reda tastera Oktavu mijenjate sa ,8 i 9

M mod služi za promjenu brzine izvođenja kompozicije tše od 30 do 188 ij od grave — do prestissimo

U ST modu (tipka C) možete mijenjati jačinu zvuka (-4 do 4) envelopu (0—7) i oktavu (1—4) za svaki kanal posebno Efekti koji se mogu postići promjenom ovih parametara su ponekad vrlo zanimljivi

Prozor Values vam daje sledeće opcije **Key signature** — služi za promjenu tonaliteta Ako pravite kompoziciju u recimo G duru onda je potrebno na početak NS staviti povisilicu (hash) na G Time će svako G u kompoziciji biti odsvirano kao Gis Želite li negdje odsvirati obično G moraćete ispred te note staviti snizilicu (znak sličan hashu) Ta snizilica će važiti samo za takt u kojem se nalazi **Time signature** — koristi se za postavljanje mjere tj koliko će taktovi da traju (od 2/2 do 16/16) **Tempo** je isto što i M mod samo što ovdje ne mijenjate brojeve već muzička naziva za tempo **Autobar** — služi za automatsko postavljanje završetka takta **Note accent** — naglašava notu a **Bar accent** naglašava prvu notu u taktu **Autosound** — odsvira svaku notu nakon ENTER **Colours** — naravno mijenjanje boja (1—8) Za šta služe **Pop up delay** i **Resolution** nisam uspio da nađem

Commands prozor ima 4 komande od kojih **Clear music** briše sve što ste napisali **Clear notepad** bi vjerovatno trebalo da obriše kanal u kojem se trenutno nalazite ali kod mene to ne radi Šta rade **Set note stem** i **Swap sound sets** ne znam

Vrlo zanimljiva opcija je **Synthesizer** Njom možete da izmijenite karakteristike svake od 7 envelopa Šta se sve može uraditi najbolje ćete vidjeti ili čuti ako pritisćete ,Z ili ? i mijenjate parametre envelope

Generalno, po prozorima se šeta pomoću kurzorskih strelica ili SPACE BAR -a (u Synthesizeru) a parametri se mijenjaju pritiskom na 8 ili 9 na numeričkoj tastaturi iz svih menija se izlazi sa ESC isto tako i reprodukcija se prekida sa ESC

Uopšte uzevši Music System je odličan program koji gotovo 100% poštuje muzičku sintaksu, rad je vrlo komforan postoji sve silne opcije koje omogućavaju da se zvuk oblikuje na hiljadu načina No ni ovaj program kao i svaki drugi nije pošteđen mana Sa **Ties** se recimo ne mogu povezati više od dvije note uzastopno Najveća zamjerka je to da možete unijeti svega 999 nota (računajući i pauze) Ja sam jedva nekako nagurao tri stranice nota To je zaista malo Dopuštam da ovo nisu sve mane programa a možda postoji još neka opcija koju nisam našao Nadam se da će se neko javiti i napisati šta rade onih par komandi koje ja nisam znao objasniti Bez obzira na sve toplo vam preporučujem Music System

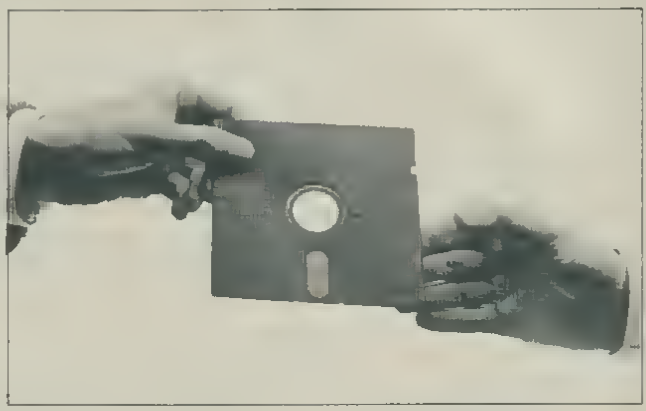
Igor Vukičević

VMS/datoteke (2)

Ne, nemoj mi prići

Pošto smo u prošlim „Računarima" upoznali osnovne svojstva operativnog sistema VMS i komunikacione metode na relacijama korisnik—računar i korisnik—korisnik, došlo je vreme da se pozabavimo datotekama i načinom na koji ih VMS skladišti i obrađuje — videćemo da se, baš kao i kod junikса, veći deo komandi VMS-a bavi upravo manipulisanjem datotekama!

Datoteke smo istini za volju implicitno koristili i u primerima iz prošlog nastavka — u toku prijavljivanja korisnika njegova prava se proveravaju analizom datoteke SYSUAF.DAT izvršavanje svake komende zahteva učitavanje raznih datoteka elektronske pošta se upisuje u specijalno pripremljene datoteke. Reč *datoteka* je, inače prilično neobične srpsko-engleske kovenica — `teka` je slani izraz za sveku dok su *data* podaci u slranoj literaturi koristi se izraz *file* (=fascikla) koji ponekad pretvorimo u domaću reč fajl. Datoteku možemo da zamislimo kao fasciklu u koju su logično poredani srodni podaci — ukoliko na primer koristimo nekog VAX-a za obradu tekste kompletan tekst koji pišamo i ispravljamo biće upisan u datoteku. Osim datoteka sa tekstom i podacima, VMS poznaje datoteke sa prevedenim i izvršnim programima (njihovim ispisivanjem na ekranu dobijaju se raznorazni efekti koji uključuju i privremenu blokadu terminala) bibliotеke sistemske datoteke kao i datoteke — direktorijume kojima ćemo uskoro posvetiti dosta pažnje.



Imena datoteka

Osnovna karakteristika datoteka je naziv koji biramo tako da asocira na namenu „zapamćenih" informacija. Naziv se sastoji od imena i tipa (ekstenzije) dok se sveka od ovih komponenata sastoji od *određenog broja* slova cifara i specijalnih znakova koji se u praksi svode na donju crtu. Frazu određenog broja smo istakli jer je ovaj broj direktno zavisen od korišćene verzije VMS-a: na verziji 3 ime se sastojalo od najviše devet znakova a tip od najviše tri slova. Na verziji 4 dopuštene su veoma dugačke imena i tipovi (i preko 30 znakova) ali je zbog lakše manipulacije i preglednijih spiskova datoteka i dalje veoma preporučljivo da ime ima najviše deset a tip tačno tri znaka.

Konstruktori VMS-a su doneli i veoma rigorozno poštovali odluku da korisnik može potpuno slobodno da bira ekstenzije — nema nikakve prepreke da izvršni program dobije „prazime" FOR a bejzik program EXE. Nema prepreke ali nema ni svrhe — deteko je boje koristiti standardne i uobičajene ekstenzije koje sistem podrazumeva jer se lako izbegavaju zabune ali i šlndi na kucanju! Najvažnije standardne ekstenzije su

EXE — izvršni program koji može da se startuje sa RUN

OBJ — prevedeni program koji treba linkovati da bi se pretvorio u izvršni na može se startovati i nema ga smisla lepiširati

LIB — bibliotеka prevedenih programa koju kao OBJ datoteke nema smisla lepiširati ni startovati

.COM — datoteka sa upravljačkim programom tj. nizom naredbi VMS-ovog upravljačkog jezika. Obzirom da se u suštini radi o datoteci sa tekstom njen sadržaj možemo da lepršamo na TYPE obzirom da se radi o nekoj vrsti programe, možemo je izvršiti (koristi se znak @ (majmun)) a na komanda RUN — više o tome docnije). Kod vlasnika PC-ja ova ekstenzija izaziva svojevrsnu zbrku — COM na VMS-u je ono što i BAT na MS DOS-u dok COM na MS DOS-u odgovara izvršnim (EXE) datotekama VMS-a!

DAT — podaci koje neki program obrađuje III proizvodi odnosno podaci od značeja za rad sistema datoteke ovoga tipa obično nema smisla lepiširati es TYPE ali se uz pomoć komende DUMP često može pogoditi smisao podataka

LIS — datoteke se izveštajem koji je proizveo neki kompajler redi se o tekstu koji se ispisuje na ekranu III štampaču a ne o izvršnom programu

LOG — „dnevnik" koji sistem proizvodi u pojedinim situacijama. Ze sistemske korisnike posebno su značajne datoteke OPERATOR.LOG koje sumiraju sve dnevne dogadaje u sistemu dok će smrtni korisnici LOG datoteke primećivati ako koriste paketnu obradu kojom ćemo se baviti nešto docnije

SYS — lip čija svrhe nije baš jasna. Asocijacije na nekakve sistemske datoteke je lepravne utoliko što se uz sam operativni sistem dobija nekoliko SYS fajlova koji bi međutim mirne duše mogli de se prezivaju DAT što je uostalom i ekstenzija nekih veoma važnih sistemskih datoteka. U jednom našem računskom centru pre par godine je zabeležena interesantna primena SYS datoteke u okviru obezbeđenja sistema posle mallh epizode sa korisnicima koji su zavlečili nos tamo gde mu nije mesto program AUTHORIZE (autorizacija novih korisnika i promena prava postojećih — više o ovom programu u završnom nepisu ove serije koji će se baviti neredbama razervisanim za privilegovane korisnike) koji je prirodno upisan u datoteku AUTHORIZE.EXE je modifikovan teko da kraire posebnu datoteku u koju će upisivati podatke o svekom korisniku koji ga je startovao kako bi osobije centra docnije ustanovilo da li ja neko bespravno vršljao po sistemu. Pošto isto to osobije cantra nlja želelo da špijunira samo sebe originalni AUTHORIZE (onaj što ne ostavlja tragove) je prepisan u susedni direktorijum i preimenovan u AUTHORIZE.SYS. ako se neko beš seli da otkuca RUN AUTHORIZE.SYS razultat je 1 0 ze njega! SYS datoteke su ukratko nešto čega običan korisnik trebe da se kloni

VMS eve u svemu podrazumeva da će korisnik izvršavati (RUN) .EXE i (@) .COM datoteke čitati (TYPE) listinge (.LIS) povezivati (LINK) prevedene (.OBJ) programe obrađivati podatke (.DAT) i izbegavati sistemske podetke (.SYS) što znači de se odgovarajuće ekstenzije u odgovarajućem kontekstu mogu izostaviti — RUN AUTHORIZE.EXE je sasvim ekvivalentno se RUN AUTHORIZE.TYPE.SYSUAF.LIS je isto što i TYPE SYSUAF.! tako dalje. Ukoliko je ekstenzija različita od podrazumevene, moraćemo da je otkucamo, na primer TYPE OPERATOR.LOG III pomenuto RUN AUTHORIZE.SYS. Vradi još reći: de razni kompajleri koje bismo mogli smatrati delom operativnog sistema podržavaju svoje ekstenzije što znači da na primer fortran bejzik i C programima odgovaraju ekstenzije FOR BAS i C. Odgovarajuća ekstenzije su rezervisane i ze kompatibilnost sa ranijim DEC-ovim računarima (PDP) i neke druge svrhe kojima se ovde nećemo baviti. Nema narevno nikakve prepreke da korisnik kraire sopstvene tipova i da na primar piemime dodeli ekstenziju „ŠPIS ili LET a datotekama koje su prenete sa PC-ja sa PC.

VMS, za razliku od MS DOS-a i junikse podržava i verzije — pun naziv datoteke se sastoji od imena ekstenzije i verzije. Korist od verzije je što se posle svekog ispravljanje nekog programa na disku čuvaju i njegove prelhodne verijenie kojima se ako se pokaže da je ispravke donele više štete nego koristi nije teško vratiti. Loša strana verzije je što se prostor na diskovima nemiloerdno troši što znači da korlenik s vremena na vreme komendom PURGE

treba da obriše stare i nepotrebne verzije svojih programa

Oznake verzije se navodi iza tipa i od njega odvaja tačkom ili tačkom i zarezom. Ukoliko se ne navede, podrazumeva se poslednja verzija (ako postoje tri verzije datoteke *proba.txt* TYPE PROBA.TXT će ispisati treću dok bi za ispisivanje druge verzije trebalo otkucati TYPE PROBA.TXT 2) pri čemu VMS kod nekih osetljivijih operacija kao što je brisanje zahteva unošenje punog naziva koji obuhvata i verziju

Osnovne radnje

Najjednostavniji način da kreiramo neku datoteku je upotreba editora EDT o kom ćemo pisati za dva meseca. Pošto nam je za upoznavanje komandi potrebna neka datoteka, kreiraćemo je kucajući

$ create proba.txt
Ova je kratka probna
datoteka sa tekstom
<CTRL><Z>
$

Kucamo, prirodno, samo istaknuti tekst ne zaboravljajući pritisak na RETURN odnosno ENTER na kraju svakoga reda. Otkucavši DIR na ekranu vidimo ime *proba.txt* zajedno sa spiskom datoteka koje je u naš direktorijum eventualno prekopirao upravnik centra kada nam je odobrio korišćenje sistema. Sa TYPE PROBA.TXT pregledamo sadržaj datoteke *proba.txt* — TYPE je, kao što se moglo i pretpostaviti, naredba koja obezbeđuje prikaz sadržaja tekstualne datoteke na ekranu. Ukoliko iza TYPE navedemo ime nepostojeće datoteke, računar će ispisati nešto poput *No such file* — verovatno se radi o grešci u kucanju!

Komandom COPY kopiramo datoteke — sintaksa je *COPY stara nova* što znači da ćemo kucajući COPY PROBA.TXT PORUKA.TXT kreirati datoteku *poruka.txt* koja je identična sa datotekom *proba.txt* (otkucajte TYPE PORUKA.TXT kako biste se u to uverili). Originalna datoteka se, međutim, ne uništava — kucajući DIR vidimo spisak datoteka u kome su mesto našle *proba.txt* i *poruka.txt*

Komanda RENAME je zadužena za promenu imena datoteke — sa *RENAME stara nova* datoteku čije je ime *stara* preimenujemo u *nova*. Početnici često zamišljaju da je naredba RENAME specijalan slučaj naredbe COPY koji po kopiranju briše datoteku čiji je sadržaj prenesen na drugo mesto. Nije međutim baš tako — sistem jednostavno modifikuje određene ukazatelje što znači da ćemo kada malo bolje upoznamo direktorijume naredbom RENAME moći da prenosimo datoteke iz jednog direktorijuma u drugi ali ne i sa jednog diska na drugi! Naredba COPY sa druge strane vrši fizičko prepisivanje sadržaja, što znači da uz njenu pomoć možemo da kopiramo podatke sa jednog medija na drugi bez ikakvih ograničenja. Ostaje još da otkucamo RENAME PORUKA.TXT PORUKA1.TXT i tako proverimo delovanje komande RENAME.

Datoteke brišemo komandom DELETE: ako zaključimo da nam datoteka *poruka1.txt* nije potrebna otkucaćemo DELETE PORUKA1.TXT * — finalni i korisnicima MS DOS-a dobro poznati džoker znak * zahteva uništenje svih verzija datoteka *poruka1.txt* dok bi izostavljanje zvezdice ograničilo delovanje komande DELETE na poslednju verziju datoteke

Stablo kataloga

Sistem koji komunicira sa više korisnika i pristupa hard diskovima velikog kapaciteta mora na neki način da reši problem razvrstavanja datoteka u grupe — ne bi se daleko stiglo ako bi sve datoteke bile na jednoj gomili, jer bi tada svaki korisnik brisao tuđe podatke i stvarao haos. Čak i sistemi koji komuniciraju sa jednim korisnikom moraju da implementiraju hijerarhijsko stablo kataloga jer bi korisnik verovatno zauvek prestao da se bavi kompjuterima kada bi, otkucavši, DIR video spisak od par hiljada datoteka!

Koren stabla kataloga se u terminologiji VMS-a veoma originalno zove 000000 i sadrži nekoliko veoma važnih datoteka i niz potkataloga — jedan od njih potkataloga bi mogao da se zove *users.dir* (DIR je standardna ekstenzija za kataloge) Potkatalog users.dir ima svoje potkataloge od kojih je jedan verovatno vaš — otkucajte SHOW DEFAULT i saznajte da se vaš katalog na primer zove [users.računari]. Kreirajmo jedan potkatalog CREATE/DIRECTORY [.PODACI]. Kucajući DIR vidimo da se u našem katalogu osim datoteke *proba.txt* nalazi i potkatalog *podaci.dir*

Kucajući SET DEFAULT [.PODACI] prelazimo u katalog koji smo upravo kreirali — SHOW DEFAULT će nas uveriti da je njegovo puno ime *[users.računari.podaci]* dok će DIR dokazati da u katalogu ne postoji ni jedna datoteka. Datoteku možemo da kreiramo primenom editora ili da je prekopiramo otkucaćemo na primer COPY [USERS.RAČUNARI]PROBA.TXT PROBA.TXT a zatim DIR — u radnom katalogu nalazi se datoteka *proba.txt* koju smo primenom prethodne naredbe prekopirali iz roditeljskog kataloga. Naredbom DIR [USERS.RAČUNARI] možemo da se uverimo da je datoteka *proba.txt* i dalje u našem osnovnom katalogu što znači da nema nikakve prepreke da datoteke u raznim katalozima imaju iste nazive: za operativni sistem je bitno da svaka datoteka ima različit pun naziv pri čemu pun naziv obuhvata i ime kataloga — dve datoteke o kojima govorimo zovu se *[users.računari]proba.txt;1* odnosno *[users.računari.podaci]proba.txt;1*

Pošto smo završili sa kopiranjem datoteka vraćamo se u osnovni katalog kucajući SET DEFAULT [USERS.RAČUNARI] ili što je u ovom slučaju potpuno isto SET DEFAULT [—] — minus označava roditeljski katalog. Zatim možemo da kreiramo druge kataloge i njihove potkataloge formirajući tako stablo datoteka i direktorijuma koje pripada samo nama

Kataloge uklanjamo primenom standardne komande DELETE — ako zaključimo da nam katalog *podaci* nije potreban obrisaćemo sve datoteke koje se u njemu nalaze primenom komande DELETE *.* a zatim otkucati SET DEFAULT [—] i onda DELETE [USERS.RAČUNARI]PODACI. VMS ne dopušta uklanjanje kataloga koji nije prazan kao i kataloga u koji smo se trenutno pozicionirali primenom komande SET DEFAULT

Sistemski katalozi

Svaki korisnik nekog VAX-a ima svoj direktorijum u koji upisuje datoteke, kreirajući po potrebi i potkataloge. Od nas zavisi da li će naše datoteke i naši katalozi biti pristupačni ostalim korisnicima sistema baš kao što i od tih ostalih korisnika zavisi da li će nam omogućiti da čitamo i menjamo njihove datoteke. Dobar deo programera otvara svoje direktorijume za čitanje jer to olakšava timski rad (ukoliko je nečiji direktorijum zatvoren sasvim je moguće da on ne zna kako da ga otvori!) ali će retko ko dopustiti drugim korisnicima centra da prepravljaju ili brišu njegova remek-dela

Osim kataloga koji pripadaju ostalim korisnicima postoji čitavo stablo takozvanih sistemskih kataloga. U njih su upisani kompajleri i interpretatori raznih programskih jezika, tekst editori i delovi operativnog sistema. Ovo stablo je veoma složeno i retko koji korisnik razmišlja o njemu: za prosečnog korisnika sasvim je dovoljno da zna da pozove one sistemske programe koji su mu potrebni. Ukoliko imate vremena možete da pokušate da pregledate deo sistemskog kataloga koji vam je dostupan — videćete da je deo sistemskih programa zabranjen za čitanje

U osnovni (000000) direktorijum pozicionirate se kucajući SET DEFAULT [000000]. Sa DIR pregledate spisak datoteka i kataloga a zatim se pozicionirate u neki od njih (SET DEFAULT ...) otkucate DIR i tako dalje. Posebno je interesantan direktorijum [SYS0] i njegovi potkatalozi [SYS0.SYSEXE] i [SYS0.SYSMGR] — prvi sadrži sve direktno izvršne sistemske programe (kompajleri, interpretatori, segmenti operativnog sistema ...) kao i datoteke SYSUAF, NETUAF, RIGHTLIST koje određuju prava svakog korisnika (ako je datoteka SYSUAF otvorena za upis svaki korisnik može da pravi haos u sistemu) dok je drugi rezervisan za upravnika sistema i sadrži procedure koje se izvršavaju kada računar počne da radi, dnevnik događaja u sistemu u vidu datoteka ACCOUNTNG i OPERATOR i tome slično. U toku pretraživanja sistemskih kataloga možete da naiđete i na direktorijum GAMES u kome su pogađate igre. Ne treba se nadati da su ove igre nekakvo svetsko čudo — najslabija akciona igra za spektrum je premija prema akcionim igrama koje se igraju na terminalu: pošto je rezolucija bedna a reagovanje na komande sporo i neravnomerno (zavisi od opterećenosti centra). Veliki sistem ipak ima i nekoliko prednosti koje će iskusan igrač brzo uočiti: avanture su izuzetno složene (na VAX-u vredi probati „Imperiju"), programi za šah mogu da budu (ali na VAX-u obično nisu) vrlo dobri i što je posebno zanimljivo akcione igre (vredi probati „Kix" i u poslednje vreme „Tetris") imaju tablicu najboljih skorova u koju se upisuju svi korisnici koji u toj igri imaju iskustva. Možete da pokušate da se ubacite na neku ovakvu tablicu ali su vam šanse slabe!

Pravo pristupa

Videli smo da nekim datotekama možemo da pristupamo a nekima ne. Što se datoteka kojima možemo da pristupamo tiče, neke možemo samo da čitamo a druge i da menjamo odnosno brišemo. Upoznajmo, dakle, naredbe kojima se ograničava pravo pristupa nekim datotekama.

Svaka datoteka pre svega ima svog vlasnika koji se u terminologiji VMS-a zove *owner* i obeležava se o — vi ste vlasnik svake datoteke koju ne bilo koji način kreirate. Na verziji 3 VMS-a jedino je vlasnik datoteke (i, naravno, upravnik centra) mogao da menja prava pristupa svojim datotekama: na verziji 4 vlasniku je data mogućnost da eksplicitno ovlasti neke druge korisnike da menjaju ova prava

Korisnici koji zajedno sa vama rade na nekom projektu su grupa (obeležavaju se sa g) dok se svi ostali korisnici centra zovu *world* i obeležavaju se w. Uvedeni su najzad i takozvani sistemski korisnici koji su sasvim logično obeleženi sa s

VMS deli pristupe datoteci na čitanje (*read* ili r) upis (*write* ili w) izvršavanje (*execute* ili e) i brisanje (*delete* ili d). Svaka datoteka ima desetak atributa koji govore ko je sve može čitati, ko izvršavati a ko menjati. Otkucaćemo DIR (OWNER) PROTECTION i pročitati nešto poput

PROBA.TXT 1 [100,3] (RWE,RWED,RWE,RE)

PROBA.TXT 1 je puno ime datoteke, [100,3] je identifikacioni kod vlasnika (takozvani UIC, *user identification code* — zavisno od toga kako su korisnici kreirani na verziji 4 VMS-a ovaj broj može da bude zamenjen tekstualnom šifrom, na primer prezimenom korisnika) dok tajanstveni znaci na kraju rada predstavljaju atribute prema shemi (System, Owner, Group, World)

U našem primeru sistemski korisnici imaju pravo da čitaju, menjaju i izvršavaju (RWE) sadržaj datoteke PROBA.TXT (datoteku sa tekstom naravno nema smisla izvršavati ali sistem o tome jednostavno ne razmišlja) vlasnik ima sva prava (RWED) koja uključuju i brisanje datoteke, korisnici koji rade na istom projektu i prema tome imaju UIC čija je prva komponenta 100 mogu da čitaju, menjaju i izvršavaju (RWE) PROBA.TXT dok ostali korisnici tekst mogu samo čitati i, ako to ima smisla, izvršavati (RE). Otkucaćemo SET PROTECTION=(W) PROBA.TXT i sa DIR /PROTECTION proveriti da li je „svetu" uskraćeno bilo kakvo pravo da pristupa datoteci. Tačnu sintaksu i primenu komande SET PROTECTION upoznaćemo za mesec dana kada ćemo se baviti i drugim bitnim komandama VMS-a

Dejan Ristanović



Titius, altius, fortius!

Već nejevijene Prve olimpijede računarskih nauka održena je u Obrazovnom centru Delte u Novoj Gorici, od 25 do 31 jula 1988 godine pod ganaralnim pokroviteljstvom Iskre Delte Computers

Sve zemlje, članice međunerodne esocijecije „Pokret nauku mladima (III International Coordinating Commitee for the presentation of science and the development of out-of-school scientific activities — I C C) pozvane su da pošalju ekipu od tri člene mlađe od dvadeset godine I njihovog vodu- Takmičari su, zatim podeljeni u grupe od tri do četiri člana tako de tekmičerska grupa ne sadrži članove samo jedne zemlje

Organizacioni odbor predložio je više zadetake sa rešenjima od kojih je međunerodne komisija odabrala dva konkursna zedatka a svaka takmičarske grupe rešavala je jedan od te dva zadetka Iete komisija ja nakon toge ocenjivale radne i dokumentovana rešenja Rezultete tekmičenja objevićemo u sladećem broju

Radni jazik olimpijada bio je angleski e dozvoljeni programski jezici Turbo Pascal i Microsoft Basic Organizator je učesnicime obezbedio lična računara sa operativnim sistemom MS DOS ili CP/M Takmičarima je praporučeno poznavanje knjige N Wirth-a Systematic Programing i Algorithms+Data Structures=Programs ili sličnih

Ne osnovu stečenih iskustava, komisija će dopuniti pravilnik takmičenja za neredne olimpijede koje treba de se održaveju svaka godina u drugoj zemlji

Učesnici eu bili dužni de sami obezbeda put, a za sve ostalo pobrinuo sa generelni eponzor Iekre DeltaComputers koji je ovim gestom pokazao de ne boluje od kratkovidosti poput mnogih domaćih proizvodeča već de gleda daleko u budućnost računarstve

procedure nosesnd
uses crt

Isključuje interni zvučnik.

function ofs(x): wo

Vraća ofset objekta *x*

function paramcount: wo

Turbo Pascal omogućava da programu budu prosleđeni parametri zadati u okviru komandne linije kojom je startovan. Ova funkcija *daje broj* tih parametara.

function paramstr (wo): st

Daje parametar broj *wo* iz komandne linije kojom je program startovan.

procedure randomize

Priprema generator slučajnih brojeva koristeći interno vreme kao *seed*

function readkey: ch
uses crt

Vraća *znak* koji je otkucan na tastaturi, ako u baferu nema *znaka* čeka se da korisnik nešto otkuca. Specijalni tasteri daju specijalna *scan* kodova.

function seg (x): wo

Vraća segmentnu adresu objekta *x*

procedure setdate (year, month, day, dayweek: word)
uses dos

Podešavanja sistemskog kalendara. *Dayweek* ima vrednost 0 6 pri čemu nula označava nedelju Ako je datum besmislen biće ignorisan

procedure setintvec (by, vector: pointer)
uses dos

Postavlje interapt vektor koji odgovara interaptu *by Vector* se obično zadaje uz pomoć operatora @ koji daje adresu interaptska procedure

procedure settime (var hour, min, sec, sec100: word)
uses dos

Podešavanje sistemskog vremena

procedure sound (wo)
uses crt

Aktivira interni generator tona *wo* određuje frekvenciju.

function sptr: wo

Vraća trenutnu vrednost registra SP

function sseg: wo

Vraća trenutnu vrednost registra SS.

function wherex: by
function wherey: by
uses crt

Vraća trenutnu x, y koordinatu kurzora.

Ovaj umetak u je u potpunosti pripremljen programom Xerox Ventura Publisher
Dizajn i grafička oprema Zoran Životić



Dejan Ristanović

Turbo Pascal 4.0

1. Uvod

Pre samo dvadesetak meseci započali smo umetak "Sa bejzika na paskal" tvrdnjom varovali ili na bejzik nije jedini programski jezik vremena u kojima je takve izjave bila vredna spomena su daleko iza nas svaki programer danes dobro zna kada je i kako programski jezik paskal nastao, kako se razvijao i zbog čega danas mnogi autoriteti smatraju da mladi programeri treba da prave prve korake baš na paskalu Znamo i da poznavanje paskala predstavlje neku vrstu ulaznice u svet profesionalne literature najposobniji svetski matematičari i programeri utrošili su natrojena časove rada da prona du ispitaju i do savršenstva poboljšaju razne algoritme da bi ih potom objavili u brojnim stručnim i naučnim knjigama i časopisima Od prezentacije u naučnim člancima teško možete da očekujete da buda prilagodena početnicima profesionalci su se dogovorili da svoja remek-dela izražavaju na mnjeviše standardizovan način koji veoma liči na paskal ako ne razumete univerzalnu notaciju, odričete se i većine usluga koja literatura može da vam pruži

Dobro su poznate i mane paskala koje se slikovito opisuju nemogućnošću prevođenja sasvim jednostavnog bejzik programa

```
10 INPUT "Kako se zoveš" A$
20 PRINT "Zdravo " A$
```

Paskal kompajler je zatim previše spor da bi na mnogim malim mašinama bio praktičan e svaki pravi haker će dodati da jezik na kome ne možete da se igrate sa PEEK i POKE nije vredan daljeg razmatranje Lepo je sva u svemu imati mašinski nezavisan jezik koji garantuje prenošenje programa sa jednog računara na drugi ali smo ipak daleko od vremena u kojima neće biti neophodno i malo mašinske zavisnosti da bi sa neki posao obavio!

Dobro standardizovani paskal bi bez sumnje stekao reputaciju jezika koji vredi znati a ne vredi koristiti de nije bilo firme Boriand čiji je Turbo Pascal premda ne baš naročito usaglašen sa standardima, naprečac osvojio milione PC-jevaca širom sveta Koje je to nove kvalitete Turbo Pascal ponudio? Pre svega komfor sveka nova verzija Turbo Pascala donosila je brzinu prevođenja izvornog programa koja je ranije bila jednostavno nezamisliva i koja je dokazala da se se jednim kompajliranim jezikom može raditi onoliko komforno koliko i se bejzi kom Sam jezik je dalje proširan toliko da podrži rad sa stringovima i datotekama, grafiku, zvuk i ukratko, sve ono bez čega jedan savremeni program ne može da opstane Turbo Pascal, najzad sjajna saradnja sa MS DOS om što mu otvara brojne perspektive u sferi sistemskog programiranja koje se na drugim mašinama retko povezuje sa paskalom

direktan pristup memoriji, pisanje interapt rutina, komunikacija sa periferijom ..

Trenutno aktuelna verzija Turbo Pascala nosi oznaku 4 0 i predstavlje blago unapređenje u odnosu na do skora veoma popularne verzije 2 i 3 kozmetičke izmena su tolike da se Turbo Pascal jednostavno ne može prepoznati ali su i suštinske novosti praćene značajnim ubrzanjima, sasvim dovoljne da se odlučite za prelazak na novu verziju Ne treba, naravno, misliti da je Turbo Pascal 4 0 program bez mana najveća zamerka je nedostatak portabilnosti što znači da jednom razvijene programe možete da koristite samo na IBM PC-ju i njegovim bližim i dalji rodacima među koje ugred budi rečeno, spada i naš TIM 011 Tu je i filozofska zamerka da je previše ekstenzija "pokupljeno" iz bejzike premda bi bilo mnogo bolje de je koncepcija rada sa stringovima kopirana od fortrana 77 Ma kakva njegove mane bile, Turbo Pascal 4 0 verovatno predstavlje najpraktičniju alatku za razvoj PC softvera i spade u one programe koji će biti zlatnim slovima upisani u istoriju i to ne samo u istoriju PC-ja nego i u istoriju računarstva uopšte Programer koji namerava da se bavi PC-jem jednostavno ne može sebi priuštiti nepoznavanje Turbo Pascala što je i za nas bio dovoljan razlog da mu posvetimo ovaj umetak.

Ograničen obim umetka nateraо nas je da se paskalom kao jezikom bavimo u minimalnoj neophodnoj meri – pretpostavka je da već poznajete standardni paskal ili da vam je pri ruci naš umetak "Sa bejzike na paskal" iz "Računara 24" što znači da ćemo se na sledećim stranicama baviti isključivo specifičnostima Turbo Pascala i njegovim vezama sa MS DOS-em

2. Instaliranje Turbo Pascala

Turbo Pascal 4.0 isporučuje se na dve ili tri diskete zavisno od toga da li ste ga nabavili samog ili zajedno sa procedurama koje podržavaju grafiku. Najvažnija je prva disketa na koju su upisane dve verzije kompajlera TURBO ja integrisani paket koji obuhvata editor kompajler i neke elemente debagera dok je TPC samostalni prevodilac koji datoteku sa izvornim programom koja je pripremljena uz pomoć bilo kog standardnog editora teksta (Norton editor IBM Profesional Editor Brief ili čak Edlin) transformiše u izvršni (EXE) oblik. U praksi ćete koristiti isključivo TURBO ali ćemo ovde potrošiti nešto prostora na opis svih datoteka koje čine paket Turbo Pascal 4.0 opis će vam pomoći da odaberete minimalan skup datoteka koje ima smisla preneti na hard disk

TURBO EXE integrisani editor i kompajler koji obezbeđuje kompletan razvoj programa.

TURBO TPL podrška komunikacije sa DOS-om štampačem, grafikom. Bez ove datoteke ne mogu se primeniti modularni mehanizmi Turbo Pascala 4.0 o kojima ćemo govoriti u jednom od sledećih poglavlja.

TURBO HLP HELP biblioteka koju koristi TURBO EXE

TINST EXE uslužni program koji obezbeđuje prilagođavanje Turbo Pascala 4.0 potrebama korisnika.

TPC EXE samostalni kompajler

TPMAP EXE uslužni program koji obezbeđuje imenanje tabela simbola koja je značajna za rad debagera.

TPUMOVER EXE - uslužni program koji omogućava prenošenja modula iz jedne TPU datoteke u drugu. Uz pomoć ovog programa može se uticati na TURBO TPL.

README COM program koji ispisuje napomene o tekućoj verziji programa sumira štamparske i druge greške u uputstvu i daje odgovor na neka standardna pitanja korisnika

README - datoteka sa tekstom koju README COM koristi.

UPGRADE EXE uslužni program koji obezbeđuje transformaciju programa pisanih na ranijim verzijama Turbo Pascala u oblik prihvatljiv za Turbo Pascal 4.0

TPCONFIG EXE kada ste pomoću programa TINST podesili razne parametre Turbo Pascala, TPCONFIG će učiniti da ist parametri važe i za samostalni kompajler TPC

MAKE EXE uslužni program koji olakšava rad sa modulima tako što pravodi samo one komponente većeg programskog sistema koje su modifikovana i komponente koje zavise od tih komponenti. Vradi ga isprobati samo oko se koristi samostalni kompajler TPC

TOUCH.COM pomoćni program koji nam omogućava da "prevarimo" MAKE i nateramo ga da prevede i neke druge datoteke

TLINK EXE Borlandov linker

GREP COM program za pretraživanje pomoću koga možemo da lociramo sva pojavljivanja nekog stringa u raznim datotekama

Na disketama se mogu naći i razne datoteke čija je ekstenzija PAS C I ASM koje zapravo predstavljaju demonstraciju raznih mogućnosti Turbo Pascala 4.0 i njegovih veza sa drugim jezicima. Ukoliko ste uz program nabavili i grafičke rutine bitne su datoteke GRAPH TPU i drajveri za razne kartice čije su ekstenzije BGI

Kada se sve sabere i oduzme vlasnici jedne disk jedinice će na radnu disketu prekopirati samo TURBO EXE..TURBO TPL, TURBO HLP i eventualno, TINST EXE Vlasnici hard diska ovome mogu da pridruže i UPGRADE EXE, TPUMOVER EXE GREP COM i neophodni drajver za grafičku karticu na primer HERC.BGI

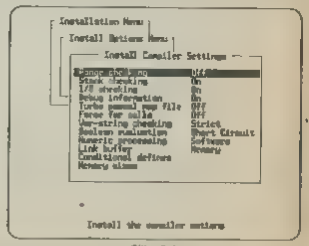
Program TINST

Program TINST kao što smo videli obezbeđuje prilagođavanje Turbo Pascala permanentnim potrebama korisnika mnoge usluge TINST-a može naravno da obavi i sam TURBO EXE ali su izmene unete u toku rada sa TURBO EXE privremene dok TINST upisuje izmene u samu datoteku TURBO EXE čineći ih tako permanentnim Posle prenošenja datoteke TINST EXE u direktorijum dodaljan Turbo Pascalu i startovanja programa na ekranu se pojavljuje meni koji nudi sledeća opcije

Compile odraduje da li ća prevedeni program biti upisan u EXE datoteku na disku ili u memoriju. Uobičajeno ja da se programi u toku razvoja upisuju u memoriju kako bi bili spremni za probno startovanje kada se razvoj završi, sam TURBO EXE će obezbediti prevođenje u nazavisni izvršni program

Options podešavanje raznih parametara. Po izboru ove opcije na ekranu se pojavljuje meni koji nudi

Compiler podešavanje parametara kompajlera (slika 2.1). Treba reći da su podrazumevane vrednosti razumno izabrane (neophodne provere granica nizova, korišćenje JUMP instrukcije minimalna neophodne dužine id) što znači da ovu opciju nema mnogo smisla koristiti osim u danima kada sistem dopunite aritmetičkim koprocesorom



Slika 2.1

procedure exec (st1, st2)
uses dos

Izvršava program koji se zove *st1* prosleđujući mu parametar *st2* Ako želite da izvršite neku DOS komandu koristite COMMAND.COM na primer exec *('command.com' '/C dir * pas')*,

procedure fillchar (x, wo, ch)

Popunjava *wo* bajtova počevši od objekta *x* znakom *ch*

procedure findfind(st, by, ver ss searchrec)
uses dos

Pronalazi prvu datoteku u direktorijumu koja zadovoljava uslove zadate džoker znacima u *st* *By* je atribut koji pokazuja da li treba tražiti datoteke sa nekim specijalnim svojstvom (bit 0 *readonly* bit 1 *hidden* bit 2 *sysfile* bit 3 *volumeid* bit 4 *directory* bit 5 *archive*) dok *ss* vraća rezultat Tip *Searchrec* je u okviru samog modula DOS deklarisan kao.

```
type
searchrec = record
              fill    array [1..12] of byte
              attr    byte
              time    longint
              size    longint
              name    string [12]
            end
```

procedure findnext (var ss.searchrec)
uses dos

Pronalazi sledeću datoteku u direktorijumu koja zadovoljava uslove zadate prethodnom primenom procedure *findfirst*

procedure getdate (var year, month, day, dayweek word)
uses dos

Očitavanje sistemskog kalendara. *Dayweek* ima vrednosti 0 6 pri čemu nula označava nedelju

procedure getintvec(by, var vector pointer)
uses dos

Daje tranutni sadržaj vektora koji odgovara interaptu *by*

procedure gettime (var hour, min sec sec100 word)
uses dos

Očitavanje sistemskog vremena.

procedure gotoxy (by1, by2)
uses crt

Pozicionira tekst kursor na (by1, by2) Gornji levi ugao ekrana ima koordinate (1 1)

function hi (ii) byte

Vraća viši bajt argumenta koji mora bai tipa *Integer* ili *word*

procedure highvideo
uses crt

Aktivira ispis istaknutim slovima.

procedure inc (x, n)

Uvećava skalarnu promenjivu bilo kog tipa za *n* ili eko se *n* izostavi za jedan. Izvršava sa brže nego *x* := *x* + 7 pa je pogodna za sistemske programa

procedure intr (by, var regs registers)
uses dos

Izvršava softverski interapt broj *by*

procedure keep (wo)
uses dos

Prekida izvršavanje programa i proglašava okončani program rezidentnim Obzirom da kompletan program ostaje u memoriji obavezno treba koristiti direktivu {$M} za određivanje minimalne neophodne veličine *heap*-a. Argument je izlazni kod koji se šalje procaduri *exit*

function keypressed: bo
uses crt

Vraća *true* ako je u baferu bar jedan znak

function lo (ii) byte

Vraća niži bajt argumenta koji mora bai tipa *Integer* ili *word*

procedure lovideo
uses crt

Aktivira ispis standardnim slovima.

function msdos (var regs.registers)
uses dos

Zahteva usluqu MS DOS-a

10. Turbo Pascal i DOS

U okviru ovog finalnog poglavlja bavićemo se vezama Turbo Pascala sa MS DOS-om tj. sticanja koje su potrebne jedino nekome ko želi da piše sistemske programe. Originalni paskal je naravno vasma daleko od toga da bude jezik za sistemsko programiranje ali su se Borlandovi programeri dosetili da dodaju procedure i funkcije koje su učinile da brojni sistemski programi u darima kada C još nije bio toliko popularan budu napisani na Turbo Pascalu. Sadržaj ovog poglavlja dakle uglavnom nema vaze se standardnim paskalom!

Prva stvar koja je neophodna za pisanje sistemskih programa je mogućnost slobodnog čeprkanja po memoriji tj. nekakav ekvivalent bejzik naredbi PEEK i POKE. Konstruktori Turbo Pascala su verovatno mogli da implementiraju istoimene procedure ali su se setili nečeg boljeg definisan je pseudo niz mem čijim se elementima pristupa sa *mem[segment:ofset]* gde su *segment* i *ofset* segmentna odnosno ofsetna adresa bajta koma se pristupa ako su indeksi promenljive one treba da budu tipa *word* Ponekad je pogodno pristupati rečima odnosno dugim rečima u kom se slučaju koriste odgovarajući pseudo nizovi *memw* i *meml* Na primer

```
mem[$0040 $0040] = 7..
reč  = memw[seg(v) ofs(v)]
duga_reč  =meml[64 3*4]
```

Hardverski orijentisani programi treba e vremena na vreme da pristupaju i portovima Primenljena metodologije je sasvim slična uvedeni su pseudo nizovi *port* i *portw*

```
portw[$20] =$2020
port[baza] ~port[baza] xor maska
while port[$62] and $80 = 0 do
```

Pisanje interapt procedura je nešto složenije. Oblik svake interapt procaduru pre svega treba da bude

```
procedure ime (flags,cs,ip,ax,bx,cx,dx,
               si,di ds,es,bp  word),
interrupt,
begin

end,
```

Interaptska procedura automatski preiauzuje sadržaj svih registara na stek i inicijalizuje DS registar što se svodi na program se slike 10 1

Očito nedostaje instrukcija CLI koja onemogućava interapte dodaćemo je sa *inline* ako smatramo da je potrebna. I porad čuvanje na steku Interaptskoj proceduri je **dopušteno** da menja vrednosti deklarisanih parametara tako da promene "prežive" kraj izvršavanja pre nego što se odlučite za promenu nekog parametra treba, jasno da dobro znate šta radite Interaptska procedura koje obrađuje hardverski generisana prekide treba da se kloni svih Turbo Pascalovih I/O procedura ili procedura za alociranje memorije pošto one nisu *reentrant* Iz istog razloga na smeju da se koriste DOS funkcije

```
PUSH  AX
PUSH  BX
PUSH  CX
PUSH  DX
PUSH  SI
PUSH  DI
PUSH  DS
PUSH  ES
PUSH  BP
MOV   BP,SP
SUB   SP,LocalSize
MOV   AX,SEG DATA
MOV   DS,AX
```

Slika 10 1

Sistemski orijentisane procedure

```
function addr (x)  pointer
```

Vraća pointer na objekt x, baš kao i operator @

```
procedure clrscr
uses crt
```

Briše sadržaj ekrana

```
function cseg  wo
```

Vraća trenutni sadržaj registra CS

```
procedure dec (x, n)
```

Umanjuje skalarnu promenljivu bilo kog tipa za *n* ili, ako se *n* izostavi za jeden izvršava se brže nego *x: =x-1* pa je pogodna za sistemske programe.

```
procedure delay (wo)
uses crt
```

Čeka *wo* milisekundi.

```
function diskfree (wo)  II
uses dos
```

Vraća broj slobodnih bajtova na drajvu *wo* Nula označava tekući drajv jedan drajv A itd.

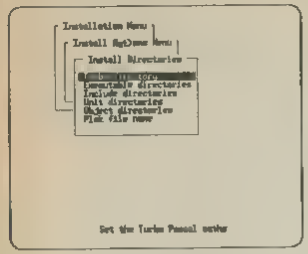
```
function diskslze (wo)  II
uses dos
```

Vraća ukupan broj bajtova na drajvu *wo* Nula označave tekući drajv jedan drajv A itd

```
function dseg  wo
```

Vraća trenutni sadržaj registra DS

Environment podešavanje parametara editora da li će pri svakom startovaiju tekuća PAS datoteka biti prekopirana u BAK, da li će se vršiti automatsko periodično snimanje da li će editor biti u modu umetanje ili modu prekucavanja koliko blankove zamenjuje tabulator I ove su vrednosti razumno izabrane što znači da ćete ih retko menjati.



Slika 2.2

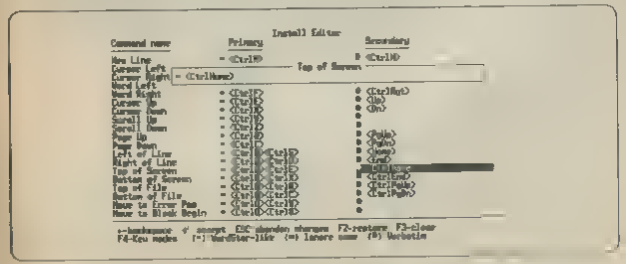
Directories određivanje lokaciga u koje se upisuju razne datoteke (slika 2 2) Za normalan rad vašem sistemu treba prilagoditi jedino Turbo *directory* koji će se na primer zvati c *jezici**pascal* Ukoliko mnogo radite sa Turbo Pascalom formira ćete razne direktorijume sa modulima u kom ćete slučaju definisati I *Include directory* odnosno *Unit directory* ostaće će polja biti prazna što znači da će sistem podrazumevati radni direktorijum. Ukoliko su datoteke nekog tipa (npr moduli) smešteni u nekoliko direktorijuma sistem omogućava navođenje njihove liste pri čemu se komponente odvajaju tačkom i zarazom baš kao da zadajete *path*

Parameters je opcije koju nema mnogo smisla instalirati. Videli smo da je krajnji cilj razvoja nekog programa EXE datoteka koju ćemo docnije startovati prostim navođenjem imena. Turbo Pascal, međutim dopušta da se iza tog imena nađu i neki parametri koji će biti prosleđeni programu ukoliko pišemo program koji sa zove DUMP EXE startovaćemo ga sa DUMP *lime* U toku testiranja program obično prevodimo I startujemo iz radnog prostora Turbo Pascala što znači da jednostavno nemamo priliku da specificiramo parametre program na bismo mogli da testiramo da se Borlandovi programeri nisu potrudili da omoguće specificiranje permanentnih parametara. Ove parametre međutim ima smisla podesiti po startovanju programa TURBO EXE e ne iz programa TINST teško bi se moglo očekivati da će svi programi koje budete razvijali imati jednake probne parametre!

Editor Commands obezbeduje izbor komandi editora koje najviše odgovaraju korisniku. Svaku naredbu editora aktivira jedna primarna (i nepromenljiva) kombinacija tastera koja podseća na WordStar i jedna sekundarna kombinacija tastera koju korisnik može podesiti u sam Turbo Pascal ugrađene su neke sekundarne kombinacije tastera koje se mogu menjati I na koje se ako poželimo možemo vratiti pritiskom na R

Promenu sekundarne kombinacije tastera vršimo teko što pozicioniramo kurzor na odgovarajući red pritisnemo ENTER I onda otkucamo željenu kombinaciju (slika 2 3) Pri ovom unošenju pomaže nam svojevrsan editor koji napuštamo se ENTER (prihvatanje izmena) II F2 (poništavanje izmena) I koji možemo ponovo da startujemo se F3 (izmene se poništavaju I istovremenđ ponovo započinju) Obzirom da su tasteri F2, F3 F4 I *backspace* zauzeti komande im dodeljujemo uz pomoć prefiksa

Pošto definišemo sve željene komande, napustamo opciju **Editor Commands** pritiskajući Esc Računar će tada proveriti da li se izmedu izabranih komandi javljaju konflikti (ako smo na primer odlučili da kombinacije tastera [HOME][UP] prenosi kurzor u prvu liniju ekrana taster [HOME] ne sme da bude samoukuna editorska komanda) I ako ih pronađe pozicionirati kurzor ne problematičnl rad I zatražiti izmenu. Ovo je značajno unapređenje u odnosu na prethodne verzije Turbo Pascala koje bi proste ispisale da konflikt postoji e onda ostavile korisniku da ga sam traži!



Slika 2 3

Display Mode obezbeđuje prilagođavanje Turbo Pascala korisničkoj grafičkoj kartici. Ukoliko ste se, poput većina Jugoslovena opremili Herkulesom izabraćete opciju *Monochrome*. Ukoliko izaberete neku od prve tri opcije računar će izvršiti test ekrana i pružiti vam priliku da proverite da li se u nekim kritičnim slučajevima javlja "sneg"

Set Colours opcija čiju upotrebu ne savetujemo. Turbo Pascal omogućava izbor boje raznih menija, prompt-ova, linija i drugih parametara radnog prostora (prepostavlja je naravno da koristite kolor monitor) - sve u svemu, gotovo 50 parametara! Upotreba ove opcije je relativno očigledna ali bi njen opis zahtevao dosta prostora u uvom umeniku obzirom da nam se čini da su inicijalna boje dobro izabrane, prostor ćemo potrošiti na korisnije stvari

Resize Windows u sledećem poglavlju videćemo da se vizuelni prikaz radnog prostora Turbo Pascala deli na dva osnovna prozora editorski i izvršni. Uz pomoć ove opcije prilagođavaju se veličine ovih prozora dok za potpuno uklanjanje izvršnog prozora treba koristiti opciju *Zoom State* iz *Options/Environment* menija.

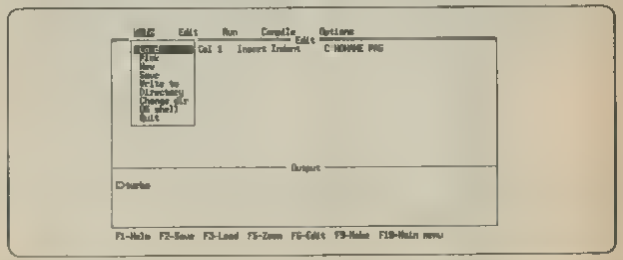
Quit/Savn pošto ste uneli željene izmene, birate ovu opciju i tako napuštate TINST. EXE. Računar će vam pružiti priliku da ignorišete izmene ili ih upišete u TURBO.EXE. Jednom upisane izmene su konačne pri čemu naravno nema mikakve smetnje da ponovo startujete TINST i poništite neke izmene ili unesete neke nove. Ukoliko se pokaže da su izmene više štetile nego koristile uvek možete da presnesete osnovnu datoteku TURBO.EXE sa originalne diskete.

3. Radni prostor Turbo Pascala

Turbo Pascal startujete kucajući TURBO [*ime*[.*tip*]] pri čemu srednje zagrade označavaju opcione argumente - ako se *tip* izostavi, podrazumeva se PAS a ako se izostavi i *ime* program će se zvati *noname.pas*. Ukoliko datoteka *ime.tip* postoji u radnom direktorijumu biće učitana i njen će se sadržaj pojaviti u editorskom prozoru. Ukoliko datoteke na postoji, računar će se pripremiti za njeno kreiranje (editorski prozor je prazan) dok će se ukoliko ima datoteke upošte nije specifično u editorskom prozoru pojaviti odgovarajuće kopirajt poruka koju okončavamo pritiskom na ENTER.

Slika 3.1 prikazuje standardni radni prostor (u terminologiji Turbo Pascala *Integrated environment*) koji se koristi za unošenje, prevođenje i testiranje paskal programa. Za početak ćemo pokušati da identifikujemo pojedine komponente radnog prostora. U samom vrhu ekrane je osnovni meni koji nudi rad sa datotekama (*File*) unošenje i ispravljanje programa (*Edit*), izvršavanje (*Run*) razne varijante prevođenja (*Compile*) i podešavanje parametara (*Options*). Slično tome poslednje linije ekrana daje osnovnu pomoć korisniku opisujući svrhu funkcijskih tastera F1, F2, F3 F5, F6, F9 i F10. Između ove dve "horizontalne menije" je radni deo ekrana podeljen u editorski prozor u okviru koga unosimo i ispravljamo program i izvršni prozor u kome se prikazuju rezultati njegovog izvršavanja. Prva linija editorskog prozora je statusna. obaveštava nas o imenu programa koji razvijamo koordinatama kursora i nekim drugim manje važnim podacima.

Pradimo na upotrebu radnog prostora počevši sa sasvim dovoljno da znaju da se pritiskom na F10 ulazi u glavni meni u kome prisicima na levu i desnu strelicu i taster ENTER biramo neku od stavki (*File, Edit, Run Compile* ili *Options*) Opcije *Edit* i *Run* nemaju nikakve podmenije iko znači da je pritisak na ENTER sasvim dovoljan da započnemo ispravljenje odnosno izvršavanje programa. Ukoliko se druge strane izaberemo neku od preostale tri mogućnosti, na ekranu će se pojaviti meni se nekoliko opcija. Po ovom meniju se kretćemo pritiskajući gornju i donju strelicu dok opciju i dalje biramo se ENTER ukoliko su potrebni još neki parametri računar će iscrtati poseban prozor za njih (slika 3.2) Ukoliko vam se učini da je ovakav rad preopr,



Slika 3.1

Write (fx, i 5) na primer upisuje u datoteku fx vrednost promenljive *i* koja zauzima tačno pet mesta ako je broj manji od 10000 dopunjava se vodećim blankovima.

procedure writeln (fx, v1, v2, .. vn)

Konvertuje podatke *v1 v2 ... vn* u ASCII oblik i upisuje ih u datoteku fx Razlika u odnosu na *write* je što se posle završenog upisivanja dodaje I *eoln* marker tj prelazi u sledeći red

Procedure i funkcije koje operišu sa netekstuelnim datotekama

function filepos (f) 8

Vraća trenutnu poziciju pointera datoteke *f* u vidu *longint*-a. Ako je pointer na početku datoteke funkcija vraća nulu.

function filesize (f) li

Vraća veličinu datoteke *f* u osnovnim jedinicama eko je datoteke netiplizirana, osnovne jedinice su blokovi čija je veličina zadata u okviru procedure koja je otvorila datoteku ako je datoteka tiptirana, veličina je broj slogova.

procedure read (ty, v1, v2, .. vn)

Učitavanje podataka *v1 v2 , vn* iz datoteke *ty* Promenljiva moraju da budu usaglašene sa tipom datoteke obično se radi o slogovima Ukoliko je pre čitanje nekog podatka *eof* bilo *true* nastupa greška

procedure seek (f, ii)

Pozicionira pointer na *ii*-ti element datoteke *f* Smatra se da je prvi element (npr slog) datoteke obeležen brojem 0 Pointer se pozicionira na sam kraj datoteke (iza poslednjeg elementa) sa *seek (f, filesize (f))*

procedure truncate (f)

Ovu proceduru izvršavamo kada želimo da uništimo podatke u datoteci i to od tekuće pozicije pointera pa do njenog kraja Jednom uništeni podaci obično se na mogu povratiti.

procedure write (ty, v1, v2, , vn)

Upis podataka *v1 v2, .. , vn* u datoteku *ty* Promenljive moraju da budu usaglašene sa tipom datoteka, obično se radi o slogovima. Pri primeni procedure **write** na datoteke koje nisu tekstualne upravljanje ispisom sa *širina:decimala* nema smisla pa prema tome nije ni omogućeno.

Procedure i funkcije koje operišu sa netipiziranim datotekama

procedure blockread (f, buf, era1, we2)

Učitava jedan ili više blokova podataka iz netipizirane datoteke *f* u bafer *buf* veličina bloka je definisana u proceduri koja je otvorila datoteku (*reset*) pri čemu je podrazumevana vrednost 128. *Wo1* određuje koliko će se blokova kopirati e *wo2* daje izveštaj o čitanju ukoliko je *wo2 = wo1* učitani su svi traženi blokovi eko je *wo2 < wo1* učitano je samo *wo2* blokova e onda je dostignut kraj datoteké.

Buf je bilo koja struktura podataka, na primer promenljiva ili (češće) niz. Podaci će se upisivati u memorijski prostor koji zauzima promenljiva *buf* eko tamo nema dovoljno mesta, sasvim je moguće da će biti uništeni neki drugi podaci što obično izaziva krah sistema.

procedure blockwrite (f, buf, we1, wo2)

Upisuje jedan ili više blokova podataka iz memorijskog bafera *buf* u netipiziranu datoteku *f*; veličina bloka je definisana u proceduri koja je otvorila datoteku (*rewrite*) pri čemu je podrazumevana vrednost 128 *Wo1* odraduje koliko će se blokova kopirati e *wo2* daje izveštaj o izvršenoj operaciji ukoliko je *wo2 = wo1* upisani su svi potrebni blokovi dok *wo2 < wo1* signalizira da je pošlo je upisano *wo2* blokova došlo do I/O greške koja se obično evodi na prepun disk

Primer 3 5 ilustruje rad se netipiziranim datotekama verujemo da ćete bez problema shvatili da se radi o brzoj *copy* naredbi

```
program kopiranje,
var  source, dest: file,
       numr: numw: word,
       buf: array [1..2048] of char,
begin
   assign (source, paramstr(1)),
   reset (source, 1),
   assign (dest, paramstr(2)),
   rewrite (dest, 1);
   writeln ('Kopira se ', filesize (source),
            ' bajtova ..'),
   repeat
      blockread  (source, buf, sizeof(buf),
                  numr),
      blockwrite (dest, buf, numr, numw),
   until (numr = 0) or (numr < > numw),
   close (source),
   close (dest),
end,
```

Slika 3 6

procedure getdir (by, st)

Promenljiva *st* dobija sadržaj koji je jednak punom imenu radnog direktorijuma. *By* je ulazni parametar koji opisuje disk 0 je radni disk, 1 je A, 2 je B, 3 C i tako dalje

function ioresult : we

Funkcija ima smisla samo ako je sa {$I-} isključeno automatsko prijavljivanje I/O greški. Tada posle svake operacije možemo da izvršimo *ioresult* i tako detektujemo grešku - nula označava uspešnu operaciju a broj različit od nule neku od grešaka

procedure mkdir (st)

Kreiranje direktorijuma čije je ime zadato sa *st*. Pri zadavanju imena treba koristiti standardnu sintaksu MS DOS-a

procedura rename (f, st)

Menja ime otvorene datoteke *f* u *st*

procedure reset (f, we)

Otvara datoteku *f* za čitanje i pozicionira pointer na njen početak. Parametar *we* je opcioni i dopušten samo kod netipiziranih datoteka pomoću njega odredujemo veličinu bloka u bajtovima (podrazumeva se 128)

procedure rewrite (f, we)

Otvara datoteku *f* za upis, uništava njen eventualni ranij sadržaj i pozicionira pointer na sam početak. Parametar *we* je opcioni i dopušten samo kod nekpiziranih datoteka pomoću njega odredujemo veličinu bloka u bajtovima (podrazumeva se 128)

procedure rmdir (st)

Uklanjanje direktorijuma čije je ime (*st*) zadato u skladu se sintaksom MS DOS-a. Da bi direktorijum mogao da se obriše u njemu se ne sme nalaziti ni jedna datoteka.

Procedure i funkcije koje operišu sa tekstualnim datotekama

procedure append (tx)

Otvara tekstualnu datoteku *tx* i priprema se za dodavanje podataka na njen kraj. Ukoliko je logičkom imenu dodaljen prazan string npr *assign (tx, '')*

pristupaće se standardnoj izlaznoj jedinici tj. ekranu monitora.

function eoln (tx) : bo

Vraća true ako su svi znaci iz tekuće linije pročitani. Tada treba primeniti *readln*

procedura flush (tx)

Primenom ove procedure obezbeđuje se da kompletan sadržaj bafere koji je DOS otvorio kako bi komunicirao se datotekom *tx* bude prepisan na disk čo znači da ni u slučaju neregularnog završetka programa neće doći do gubitka podataka.

procedure read (tx, v1, v2, ... vn)

Učitava i konvertuje podatke iz tekstualne datoteke *tx*. Svaki podatkk mora biti kodiran u ASCII obliku i to u skladu se tipom promenljiva (*integer, real, Boolean char* ili *string*) Ako se dostigne kraj rada procedure vraća CR (chr(13)) ako se dostigne kraj datoteka, vraća Ctrl Z (chr(26))

procedure readln (tx, v1, v2, ... vn)

Učitava i konvertuje podatke iz tekstualne datoteke *tx* kao i *read*. Razlika je jedino u tome što se po završenom čitanju preskače *eoln* marker i prelazi u sledeći red

function seekeof (tx) : be

Vraća *eof* status tekstualne datoteke. Funkcija je identična se *eof* ali preskače eventualna tabove blankove i prazne linije na kraju datoteke.

function seekeoln (tx), be

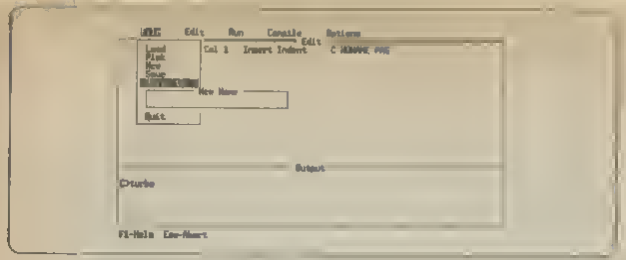
Vraća *eoln* status tekstualne datoteke. Funkcija je identična se *eoln* ali preskače eventualne tabove i blanko simbole na kraju reda.

procedure setvuxbuf (tx, buf, we)

Svakoj tekstualnoj datoteci interno se dodeljuje bafer od 128 bajta koma pristupaju procedure *read* i *write*. Za neka aplikacije veličina ovog bafera može da bude ograničenje pa je predvidena mogućnost njegovog produženja ili skraćenja posle ove naredbe datoteka *tx* će koristiti bafer *buf* (obično je *buf* neki niz) čija je dimenzija *we*. Procedure *settextbuf* obavezne treba izvršiti pre otvaranja datoteka tj. izmedu *assign* i *reset/rewrite*

procedure write (tx, v1, v2, , vn)

Konvertuje podatke *v1, v2, ... vn* u ASCII oblik i upisuje ih u datoteku *tx*. Svi podaci moraju da budu tipa *integer, real, Boolean char* ili *string* i njihovim se formatom može upravljati sa *pozicija decimala*



Slika 3.2

stavku menija možete da izaberete pritiskom na taster sa odgovarajućim slovom opciju *Write to* sa slike 3.2, na primer litrate pritiskom na W u kom slučaju računar ne zatraži dodatni pritisak ne ENTER.

Pošto smo upoznali osnove, možemo da se potrudimo i oko komfora. pritisak na F10 i pomeranje kurzora po raznim opcijama glavnog menija svakako nije operacija koja se obavlja previše brzo Zato su konstruktori Turbo Pascala predvideli brojne "prečice" nazvane *hotkeys* specijalno definisani tasteri odnosno parovi tastera omogućavaju direktan izbor neka od važnih stavki menija u skladu sa slede čim spiskom

Taster	Funkcija
F1	Pristup HELP bibliotesi
F2	Snimanje program na disk pod tekućim imenom
F3	Učitavanje programa sa diska
F5	Zumiranje aktivnog prozora na ceo ekran/kombinovanje prozora
F6	Izbor editorskog/izlaznog prozora
F9	Make
F10	Glavni meni
Alt F1	Prethodni HELP
Alt F3	Pick
Alt F5	Prikaz izvršnog ekrana
Alt F9	Prevodenje programa
Alt F10	Prikaz kopirajt poruke
Alt C	Ulazak u *Compile* meni
Alt E	Ulazak u editor (*Edit*)
Alt F	Ulazak u *File* meni
Alt O	Ulazak u *Options* meni
Alt R	Izvršavanje programa (*Run*)
Alt X	Povratak u DOS
Ctrl F6	Sledeći prozor

Kako ovi "vrući tasteri" deluju? Nije teško umesto da pritisnemo F10 i tako udemo u glavni meni pa da onda pomeramo kurzor do opcije *Run* i pritiskamo ENTER pritisnućemo Alt R Sasvim slično tome pritiska na F2 zamenjuje F10/File/Save ili Alt F S.

Ostalo je još da razmislimo klupka zvano prozori. Turbo Pascal podržava editorski (*edit*) i izvršni (*output*) prozor: u prvom unosimo i ispravljamo program a u drugom posmatramo rezultate njegovog izvršavanja Pomeranje izmedu editorskog i izvršnog prozora obavljamo uzastopnim pritiscima na F6 ako je ekran podeljen na dva dela, dupla linija nas podseća na prozor koji je aktivan i čiji će sadržaj skrolovati ako pritiskamo tastere [UP] i [DOWN] Ukoliko je izvršni prozor uklonjen se ekrana (*Options/Environment* meni opcija *Zoom windows*) pritiskom na F6 ceo ekran se dodeljuje izvršnom prozoru (sadržaj ćak može i da skroluje pritiskte taster Ctrl i računar će ispisati kako) što znači da listing programa nećemo videti dok ponovo na pritisnemo F6 Turbo Pascal, najzad omogućava i prikaz kompletnog izvršnog ekrana bez ikakvih linija ili drugih elemenata radnog prostora dovoljno je pritisnuti Alt F5

HELP biblioteka

Poput svakog ozbiljnijeg programa današnjice Turbo Pascal 4.0 je dopunjen veoma kompletnom HELP bibliotekom koja ukida potrabu sa stalnim listanjem uputstva uputstvo naravno, i dalje treba pročitati ali ća odgovore na mnoga svakodnevna pitanja pružati upravo HELP pa je veoma važno da u samom početku naučite da koristite njegove usluge. A korišćenje je sasvim jednostavno kada je kurzor pozicioniran na neku od opcija nekog od menija treba pritisnuti F1 i na ekranu će se pojaviti manji ili više kompletan opis te opcije ako čitav opis ne može da stane u prozor ostatak pregledate pritiskima na PgUp i PgDn. Pritiskom na Esc napuštate HELP i vraćate se opciji koju ste proučavali.

Ukoliko se u okviru HELP prozora pojavi neki istaknuti (na monohrom monitorima belé na kolor monitorima ciarbani) tekst računar vam pruža mogućnost de se raspitate o nekoj srodnoj temi treba samo pozicionirati kurzor na istaknuti segment i pritisnuti ENTER U okviru novog HELP prozora mogu se naći ukazatelji na neke treće oblasti i tako do prilične dubina na prethodni HELP ekran vra ćamo se pritiskom na Alt F1

Obezbeđen je nalazi t pregled kompletne HELP biblioteke dva puta pritiskeme F1 I ne ekranu se pojavljuju sve teme HELP-a koje, jasno ne mogu da stanu u jedan prozor pritiscima ne PgDn vidimo ostatak Pozicioniranjem kurzora ne neku od teme I pritiskom na ENTER dobijamo detaljne obaveštenje iz izabrane oblasti

Ostatak ovoga poglavlja posvećujemo relativno detaljnom opisu svih stavki raznih menija radnog prostora Turbo Pascala verujemo da ćete za sada samo proteteti pogledom preko neoko nevezanih naredbi ali će u docnijem praktičnom radu tekst poslužiti kao dragocen zapimi HELP

Struktura menija

File meni (ti Alt F) obezbeduje rad se datotekama – unošenje I animanje programa, pregled radnog direktorijuma I privremeno odnosno konačne napuštanje Turbo Pascala

Load (ti F3) učitava paskal program u memoriju Polito se tekući program nepovratno briše Turbo Pascal će proveriti da ti su unesene neke izmene pa će vam, ako jesu, pružiti poslednju priliku da program snimite na disk

Pick (ti Alt F3) učitavanje paskal programa po kratkom postupku Kada budemo upoznali modularni princip programiranja razumećemo de se ist program može prostirati u nekoliko datoteka koje treba neizmenično obrađivati Zato Turbo Pascal čuva imena prethodno učitavanih datoteka I obezbeduje izbor neke od njih prostim pomeranjem kurzora. Polito se opcijom *Pick* tekući program nepovratno briše, Turbo Pascal će proveriti da li su unesene neke izmene pa će vam ako jesu pružiti poslednju priliku da program snimite na disk

New briše tekući program i priprema se za kucanje novog koji se (za sada) zove NONAME PAS Polito je brisanje tekućeg programa nepovratno Turbo Pascal će proveriti da li su unesene neke izmene pa će vam ako jesu, pružiti poslednju priliku da program snimite na disk

Save (ti F2) snimanje radnog programa ne disk pod tekućim imenom. Uobičajene mere predostrožnost Turbo Pascala se ovde ne primenjuju što znači da će snimanjem biti nepovratno i bez upozorenja uništen ranij sadržaj istoimene datoteke pa treba biti vrlo oprezan Ime je uvek prikazano u desnom uglu statusne linije editora.

Write to snimanje programa pod nekim drugim imenom Korisnik unosi novo ime u odgovarajuće polje novog prozora e računar proglašava to ime za radno što znači da će budući pritisci ne F2 snimati program pod novim imenom

Directory spisak datoteka u radnom direktorijumu. Korisniku sa prethodno pruža mogućnost da unese "masku" koja će izdvojiti samo neke datoteke koriste se uobičajeni džoker znaci DOS-a što znači da, na primer maska * PAS daje spisak svih paskal programa u radnom katalogu.

Change dir promena radnog kataloga koja obuhvata i eventualnu promenu radnog diska.

OS shell privremeno napuštanje Turbo Pascala i prelazak u MS DOS U Turbo Pascal se vraćamo kucajući EXIT pri čemu sa status radnog prostora i sam paskal program u memoriji ne menjaju

Quit napuštanje Turbo Pascala i povratak u MS DOS. Polito se na ovaj način nepovratno uništava program u memoriji Turbo Pascal će proveriti da li su unesene neke izmene pa će vam ako jesu pružiti poslednju priliku da program snimite na disk.

Edit komenda (ti Alt E) izaziva prelazak u editor koji obezbeduje unošenje I ispravljanje paskal programa. Editorom ćemo se baviti u okviru sledećeg poglavlja.

Run komenda (ti Alt R) izaziva prevođenje i startovanje tekućeg programa. Ukoliko sa u toku prevođenje ukaža potreba za nekim drugim datotekama (npr tipa PAS ili TPU) one će automatski biti učitane i uključene u izvršni program Neposredno pra startovanja programa prelazi se u izvršni ekran dok se po okončanom izvršavanju ispisuje *Press any key to return to Turbo Pascal* i čeka. Izuzetak su greške kojima ćemo se nešto docnije detaljno baviti.

Compile meni (ti Alt C) obezbeduje razne varijante prevođenja programa Prevođenje se kao što smo videli implicitno izvršava I komandom **Run** što znači da u toku razvoja programa **Compile meni** nije naročito potreban njegovu pravu vrednost upo znaćemo tek kada zažalimo da krairamo nezavisan EXE program ili da povežemo komponente nekog većeg programskog paketa

Compile prevođenje programa I (ukoliko je izabrana opcija *Destination disk*) upis izvršne datoteke na disk

Make (ti F9) u jednom od sledećih poglavlja videćemo da se veći programski paket može sastojati od raznih modula pri čemu je svaki modul upisan u odvojenu datoteku. Komandom *Make* obezbeđujemo kompajliranje svih datoteka koje su menjane polito su prethodni put prevedene (računar konsultuje datum i vreme kreiranja PAS I TPU datoteke što znači da je ispravno fuukcionisanje česovnike veoma bitnoj – Turbo Pascal je dovoljno inteligentan da prevede I module koji pozivaju modifikovene programe Komandu *Make* dakle treba izvršiti pra komande **Run** ako su vršene intervencije na nekim nezavisnim modulima.

Build isto što i *Make* ali se zahtsve prevođenje svih modula nekog programskog sisteme bez obzira da li su modifikovani posle prethodnog prevođenja ili ne. Komandu ćemo dakle izvršiti ukoliko sistem koji razvijamo pretrpi neke veće izmene u strukturi podataka ili ukoliko nismo sigumi da nam je časovnik uvek bio doteran. Uzavši u obzir brzinu Turbo Pascala komandu *Make* ne moramo ni da koristimo pri razvoju programskih paketa srednje veličine!

Destination određuje gde se prevedeni program smešta. Odredište može biti memorija (program se tada može izvršiti sa **Run**) ili EXE datoteka na disku koja je nezavisna od Turbo Pascale Bez obzira na izabrano odredište prevedeni modul se uvek upisuju u TPU datoteke na disku

Find error ukoliko prl izvršavanju programa dođe do greške (npr deljenje nulom prekoračenje opsega ili nedefinisana funckija) kontrole se automatski vraća editoru kurzor se pozicionira na liniju izvornog koda u kojoj je greške nastupila I u statusnoj liniji se ispisuje propratni tekst. Ukoilko je medutim, do greške došlo prilikom izvršavanja neza visne EXE datoteke (EXE datoteka bi dodusa tre balo kreirati tek kada je razvoj programa završen ali

Samo se po sebi razume da *fizičko ime* može da sadrži ne samo ime datoteke nego I ime diske odnosno direktorijuma na primer *assign (probna, c \razvoj\paske\proba dat)* dopušteno je ćak I korišćenje string promenljive što znači da se tasta ture možamo učitati ime datoteke I onda na neki način obraditi njen sadržaj

Samo povezivanje logičkog i fizičkog imena nije dovoljno datoteku treba otvoriti U standardnom paskalu sve su datoteke sekvencijalne što znači da su potrebna samo procedura *reset (ime)* I *rewrite (ime)* prva otvara datoteku za čitanje a druga za upis pri čemu sa podrazumeva da pre otvaranja za upis raniji sadržaj treba uništiti Ukoilko pokušate da otvorite nepostojeću datoteku za čitanje biće prijavljena greška koju treba na vreme ukinuti ugrađivanjem opcije {$I-} u odgovarajući segment koda (pogledajte peto poglavlje posle otvaranje koristi sa funkcija *ioresult* koja vrednošću različitom od nule signalizira postojanje greške i pruža priliku programu da ispiše odgovarajuću poruku ili preduzme neke druga akcije.

```
program sumafajle (output),

type    rfile =   file of real
var     f:        rfile,
        fname:    string

procedure sumfile(var datafile: rfile),
var   suma:  real,
      x:     real,

begin
   reset (datafile),
   suma := 0;
   while not eof (datafile) do
   begin
      read (datafile, x),
      suma := suma+x
   end,
   close (datafile),
   writeln ('Suma je ', suma: 10 5),
end,

begin
   repeat
      write ('Ime datoteke: '),
      readln (fname),
      if fname = '' then halt,
      assign (f, fname),
      sumfile (f),
   until false
end
```

Slika 9.5

Primer 9.5 ilustruju gotove sve o čemu smo govorili učitavanju sa racionalni podaci iz datoteke a zatim sa na ekranu izdaje njihov zbir Primetimo da je za "sumiranje datoteke" korišćena procedura I da je čitava datoteka prenesena kao parametar paskal sehteva da datoteka bude prenesena po imenu (sa *var*) ćak I eko njen sadržaj neće biti promenjen. Polito je rad sa datotekom završen zatvaramo je sa *close*

Što se ekstenzija Turbo Pascale tiče stvari su prilično složene. Postoje pre svega, dve vrste datoteka tipizirane i netipizirane dok se tipizirane datoteke dele na tekstualne i "ostale" datoteke su najbliže standardnom paskalu upisuju se binarni podaci pristup je najčešće sekvencijalan I uveriin je jedino novitat u vidu procedura *append* pomoću koje otvaramo datoteku na čijl se kraj dopisuju novi podaci Netipizirane (*untyped*) datoteke su specifična novost deklarišemo ih jednostavnim *ime file* i prema potrabi im pristupamo sekvencijalno II relativno Osnovna jedinica je ovde blok koji iz istorijskih razloga. ima 128 bajta u praksi ćemo koristiti iskljućivo blokove veličine jednog bajta jer veličinu datoteke neposnatog tipa možemo da izrazimo jedino u bajtovime Netipizirane datoteka je, kratko rečeno, datoteka koju na neki način obrađujemo ne pitajući se mnogo šta njen sadržaj predstavlja.

Ostatak ovog poglavlja posvećujemo funkcijama I procedurama koja obezbeđuju rad se datotekame različitih tipova. Uz konvencije iztičene u prethodnom poglavlju koristićemo I.

tx	tekstualna datoteka
ty	datoteka odredenog tipa
nt	datoteka bez odredenog tipa
f	bilo koja datoteka

Procedure i funkcije koje operišu se svim datotekama

procedure assign (f, st)

Logičko ime *f* se povezuje se datotekom ne disku *st* Ukoliko je *st* prazan string *f* će ubuduće biti konzole (ekran odnosno tastatura) ali ne specijalan način koji podržava redirekcija MS DOS-a.

procedure chdir (st)

Menja radni direktorijum u *st* Važi MS DOS sintakse.

procedure close (f)

Zatvaranje datoteke. Ukoilko su u datoteku upisivani neki podeci *se close* zahtevamo da se kompletan sadržaj bafera koji je DOS koristio prenese ne disk Datoteke treba obavezno zatvarati jer neregularan kraj izvršavanja programa može da dovede do gubitka dela podataka u nezatvorenim datotekama.

function eof (f): bo

Vraća *true* eko je marker pozicioniran na kraj datoteke *f* tj eko je dalje čitanje besprednetno

procedure erase (f)

Brisanje datoteke *f* koja prethodno mora biti otvorena.

```
type mes = (januar, februar, mart, aprit,
            maj, jun, jul, avgust, septembar,
            oktobar, novembar, decembar),
     datum = record
               dan     1..31,
               mesec   mes,
               godina  1850..1988,
             end,
     radnik = record
               ime     string [ime_duz],
               prezime string [prez_duz],
               pol     (muskarac, zena),
               rodjen  datum,
               staz:   0..40,
               satl    0..300,
               plata   real
             end,

var  zaposleni  : array [1..max_zap] of radnik,
     unesent    : radnik;

begin
   zaposleni[3].rodjen.dan := 12;
   zaposleni[3].rodjen.mesec := mart;
   zaposleni[3].rodjen.godina := 1952;
   { ... }
end
```

Slika 9.2

```
begin
   with zaposleni[3].rodjen do
   begin
      dan := 12;
      mesec := mart;
      godina := 1952
   end;
   { ... }
end
```

Slika 9.3

vaju pristup odgovarajućim poljima. "Polja u poljima" očito čine izraze veoma dugačkim što je pre svega, neprijatno za kucanje e zatim i nepregledno U primeru 9.3 iskoristili smo naredbu *with* na početku napišemo *with ime_sloga do* e zatim navodimo samo imena polja kojima pristupamo Struktura *with* za sesvim očekivano zaključuje se *end*

Navoja je što podaci u praksi nisu leti za svaki slog Uzmimo, na primer da u našem programu za evidenciju radnika treba uneti i podatke o rešenom stambenom pitanju. Ukoliko je neki radnik rešio stambeno pitanje treba nam (eventualno) podatak o tome da li mu je dodeljen društveni stan i (eko jeste) koje je godine to bilo Ukoliko, sa druge strane radnik nije rešio stambeno pitanje treba nam podatak o tome da li već ima nekakav smeštaj i koliko poena ima na stambenoj listi U zavisnosti od polja *stambeno pitanje* dalde ostatak sloga može de ima dve različite strukture. Paskal začudo omogućava čak i ovakvu vratolomiju što dokazuje primer 9.4

```
type
   lzbor  = (reseno,nije_reseno),
   radnik = record
               ime     string [20]
               prezime string [30],
               pol     (muskarac, zena),
               staz:   0..40,
               satl    0..300,
               plata   real
               caza stambeno_pitanje: lzbor of
               reseno
               (
                  stan    Boolean,
                  godina. Integer
               ),
               nije_reseno
               (
                  smestaj Boolean,
                  poena   Integer
               )
            end,
```

Slika 9.4

Datoteke

Slogovi u memoriji mogu da budu interesantni i korisni ali oni Ipak nisu "ono pravo" slog uvek povezujemo sa datotekom ne nekom mediju spoljne memorije obično disku.

Naredba *write (x,y)* ispisuje vrednosti *x* i *y* na ekran tj u datoteku *output* koja se uvek podrazumeva, mogli smo dakle da napišemo i *write (output x, y)* Umesto datoteke *input* ili *output* možemo da koristimo i druga datoteke koja ćemo jasno najpre deklarisati.

Deklaracija datoteke je unekoliko slična deklaraciji metrica *lime file of tip* Vrle je bitno da razlikujemo dva tipa datoteka *file of char* i *file of bilo_šta_drugo* *File of char* se zove tekstualna datoteka (deklarišе se sa *ime text* e ne *ime file of text*) i sasvim je ravnopravan se datotekama *input* i *output* što znači da njagov sadržaj možemo da kreramo i modifikujemo primenom običnog tekst editora. Sve druge datoteke nazivamo *binarnim* njihov sadržaj može da se kreira I čita pomoću paskal programa ali ne I pomoću editora – dampovanjem binarne datoteke zaključujemo vrlo malo o njenom sadržaju! Binarne datoteke ipak imaju jednu veliku prednost. dok u tekst datoteku možemo da upisujemo sama cela racionalne I logičke veličine I stringove, u binarni faji možemo da prenesemo promenljive specijalnih tipova koje smo sami definisali.

Imena datoteka koja smo do sada pominjali su *logička* tj imaju značaj jedino u okviru samog programa Svako logičko ime treba povezati se konkretnom datotekom na disku za šta se u standardnom paskalu koristi deklaracija program U Turbo Pascalu ovo je iz osnova promenljeno deklaracijom *program* se **ne** može uspostaviti veza izmedu logičkog I fizičkog imena neke datoteke pa treba koristiti specijalnu procaduru *assign* koja se poziva se

assign (logičko_ime fizičke ime)

svaki netrivijalan program ima bagi računar će ispisati jedino segmentnu I ofsetnu adresu greške Tada ćemo aktivirati TURBO učitati I prevesti program (odradište treba da bude disk) Izabrali opciju *Find error* I uneti adresa koja je računar ispisao u formatu SSSS OOOO Kurzor će se za trenutak naći na problematičnoj liniji koju kada otklanimo u čemu je stvar možemo da ispravimo.

Primary file kada se radi se više modula opcija *Make* odnosno *Build* mora da "zna" koji je od njih glavni Ukoliko uz pomoć ova opcije na zadamo ime glavnog programa, kompajler će smatrati da je glavni onaj program koji je trenutno u memoriji

Get info razna obaveštenja o radnom programu ovu ćemo opciju najčešće koristiti kako bismo stekli utisak o raspoloživoj memoriji.

Options menl (ili Alt O) obezbeduje podešavanje raznih parametara radnog prostora. Najvažnije je shvatiti da parametri podešeni u okviru **Options** menija imaju prednost nad parametrima koji su podešeni programom INSTAL što je sasvim prirodno

INSTAL koristimo da odredimo vrednosti koje su nam stalno potrebne dok pomoću **Options** menija prilagodavamo radni prostor nekoj specifičnoj aplikaciji.

Compiler: podešavanje parametara kompajlera sumirarih na slici 2 1 Podrazumevane vrednosti su dobro izabrane ali ćemo ipak ponekad poželeti da aktiviramo *Range Checking* (produžava kod ali obezbeduje da neko a[i] =0 na upiše nulu na mesto koje će izazvati krah programa Vili računara)

Environment: uz pomoć ove opcije definišemo razne parametre editora na način opisan u prethodnom poglavlju

Directories uz pomoć ove opcije definišemo direktorijume Turbo Pascala na način opisan u prethodnom poglavlju

Parameters krajnji cilj razvoja nekog programa je EXE datoteka koju ćemo docnije startovati prostim navođenjem imena Turbo Pascal međutim dopušta da se iza tog imena nadu i neki parametri koji će biti prosleđeni programu ukoliko pišemo program koji se zove PRNT.EXE startovaćemo ga sa *PRNT ime* U toku testiranja program obično ora. vešimo i startujemo iz radnog prostora Turbo Pascala što znači da jednostavno nemamo pritiku da specificiramo parametre program na bismo mogli da testiramo da se Borlandovi programeri nisu potrudili da omoguće specificiranje permanentnih parametara Ako tih parametara ima više razdvajamo ih blanko simbolima.

Save options uz pomoć ove naredbe snimamo već podešene opcije na disk pod imenom TURBO.TP Ukoliko Turbo Pascal po startovanju pronađe ovakvu datoteku u radnom direktorijumu automatski će je učitati što znači da po svakom startovanju nećemo ponovo kucati komandne opcije imena radnih direktorijuma, parametre editora.

Load options učitavanje sadržaja datoteka TURBO.TP koja je snimljena primenom prethodne opcije Datoteke se automatski učitava po startovanju Turbo Pascala što znači da ćeme *Load options* koristiti razmerno retko.

4. Upotreba editora

Editor ugrađen u Turbo Pascal 4 0 obezbeduje brzo I jednostavno unošenje I ispravljanje programa na paskalu ili čak nekom drugom programskom jeziku radi se o tipičnog programerski orijentisanom editoru koji zahteva pamćenje većeg broja komandi ali zato do željenih rezultata dovodi veoma brzo

Editor aktiviramo izborom stavke *Edit* iz glavnog menija ili mnogo češće pritiskom na Alt E Tekst unosimo na najstandardnijl mogući način pri čemu na kraju svakoga rada obavezno pritiskamo ENTER Sam editor u izvesnoj med podržava nazubljivanje programa što znači da će po pritisku na ENTER kurzor umesto na početak sledećeg reda, biti poslavljen ispod prvog znaka u prethodnom redu (ova se mogućnost zove *Autoindent* I može se po potrebi isključiti) Za nazubljivanje je zgodan I tabulator koji predstavlja zamenu za nekolilko blanko simbola broj simbola se podešava uz pomoć opcije *Environment* iz **Options** menija.

Editor napuštamo pritiskom na F10 (povratak u glavni meni) mada nema nikakve prepreke da pritisnemo Alt R Alt C ili čak Alt X I tako direktno izabe remo neku od opcija glavnog menija.

Pomeranje kurzora

Znak levo	Ctrl S ili [LEFT]
Znak desno	Ctrl D ili [RIGHT]
Reč levo	Ctrl A ili Ctrl [LEFT]
Reč desno	Ctrl F ili Ctrl [RIGHT]
Linija gore	Ctrl E ili [UP]
Linija dole	Ctrl X ili [DOWN]
Skrol na gore	Ctrl W
Skrol na dole	Ctrl Z
Stranica gore	Ctrl R ili [PgUp]
Stranica dole	Ctrl C ili [PgDn]

Ubrzano pomeranje kurzora

Početak reda	Ctrl Q S ili [Home]
Kraj reda	Ctrl Q D ili [End]
Vrh prozora	Ctrl Q E ili Ctrl [Home]
Dno prozora	Ctrl Q X ili Ctrl [End]
Početak programa	Ctrl Q R ili Ctrl [PgUp]
Kraj programa	Ctrl Q C ili Ctrl [PgDn]
Početak bloka	Ctrl Q ili
Kraj bloka	Ctrl Q K
Prethodni položaj	Ctrl Q P
Pozicija greške	Ctrl Q W

Komande editora

Umetanje i brisanje teksta		Ostale komande	
Insert / Overtype	Ctrl V ili [Ins]	Prekid operacije	Ctrl U
Ubacivanje linije	Ctrl N	Autoindent on/off	Ctrl O I ili Ctrl Q I
Brisanje linije	Ctrl Y	Kontrolni prefiks	Ctrl P
Brisanje do kraja linije	Ctrl Q Y	Par zagrada napred	Ctrl O [
Brisanje prethod znake	Ctrl H ili [Backspace]	Par zagrada nazad	Ctrl O]
Brisanje znaka	Ctrl D ili [Del]		
Brisanje reči	Ctrl T	Traženje stringa	Ctrl O F
		Traženje sledećeg	Ctrl L
Rad sa blokovima		Traženje i zamena	Ctrl O A
Markiranje početka	Ctrl K B ili F7	Postavljanje markera n	Ctrl K n
Markiranje kraja	Ctrl K K ili F8	Pozicija markera n	Ctrl Q n
Markiranje reči	Ctrl K T	Povratak u glavni meni	F10
Štampanje bloka	Ctrl K P	Učitavanje programa	F3
Kopiranje bloka	Ctrl K C	Ignorisanje izmena	Ctrl K D ili Ctrl K O
Brisanje bloka	Ctrl K Y	Uvažavanje izmena	Ctrl K S ili F2
Skrivanje bloka	Ctrl K H	Vraćanje obrisane linije	Ctrl O L
Pomeranje bloka	Ctrl K V		
Učitavanje bloka	Ctrl K R	Tab	Ctrl I ili [TAB]
Upisivanje bloka	Ctrl K W	Tab mod	Ctrl Q T ili Ctrl Q T

Komande editora

Editor u praksi

O upotrebi editom gotovo da nema šta da se kaže svaki korisnik koji je bilo kada radio sa nekim tasst procesorom će brzo naučiti da pomera kurzor umeće i briše tekst, razdvaja i spaja redove (kurzor se pozicionira na mesto razdvajenja odnosno spaja nje pe se pritisna ENTER odnosno [DEL]) i obavlje slične operacije. Mogućnost promene bilo koje edi torske komande opisane u prethodnom poglavlju dobro dođe svakom korisniku koji je već navikao na komande nekog editora.

Operacije se većim blokovima teksta zasnivaju se na tasterima F7 i F8 pomoću kojih se markira početak odnosno kraj bloka Prostor između početka i kraja bloka je na ekranu istaknut inverznim ili obojenim slovima i na raspolaganju nam je njegovo brisanje kopiranje pomeranje štampanje snimanje na disk. Po završetku raznih operacije blok ostaje istaknut što je potencijalna smetnja isticanje poništavemo tako što marker kraja postavimo pre markera početka odnosno pritiskom na Ctrl K H

Obezbeđena je i naredba za pretraživanje FIND (Ctrl Q F) koju aventualno dopunjavamo izborom opcija. String koji tražimo može da bude dugačak najviše 30 znakova kontrolni karakteri se unose uz pomoć prefiksa Ctrl P kraj rada tražimo se Ctrl M J dok ulogu džoker znaka ima Ctrl A. Pošto zadamo string pruža nam se prilika de ga dopunimo opcija ma

B pretražuje program unatrag tj od pozicije kur zora prema početku datoteke

G pretražuje čitav tekst bez obzira na trenutnu poziciju kurzora.

L pretražuje samo markirani blok

n (bilo koji broj) pronalazi *n*-to polavljivanje stringa

U ignoriše razlike između velikih i malih slova

W traži isključivo reči što bi značilo da tekst *ait* neće biti pronađen u reči *halt*

Opcije se kada god to ima smisle mogu kombinovati što znači da opcija GU (ili UG) izaziva pret raživanje kompletnog teksta uz ignorisanje razlika između velikih i malih slova.

Kada traženi string bude pronađen možamo de otkucamo Ctrl L i tako nastavimo pretraživanje ili da se Ctrl O F zadamo novi string koji može nastati i editovanjera prethodnog.

5. Upotreba kompajlera

Kompajler ugrađen u radni prostor Turbo Pas cala zaista nije teško upotrebiti otkucate Alt R i program je prevedeni S vremena na vreme se, međutim ukazuje potreba za nekim "specijalnim efektima" koje postižemo uz pomoć takozvanih opcija kompajlera Opcije ili direktive Turbo Pascal kompajlera dele se na prekidačke parametarske i kondicionalne.

Prekidačke direktive

Prekidačke direktive uključuju ili isključuju neke od karakteristika kompajlera Zadaju se u vidu komentara koji počinje znakom $ iza koga je sme štena sama opcija u vidu jednog slova i parametra + (opcija aktivna) odnosno (opcija isključena) Svakt prekidač ima neku podrazumevanu vrednost

segment stringa *st1* od *i*1-og znaka do kraje Primer e = *'Pascal b = copy (e,3,2)* daje vrednost sc

procedure delete (st, i1, i2)

Uklanjanje *i2* znakova iz stringa *st* i to počevši od znaka čija je pozicija *i1*. Ukoliko *i1* pokazuje iza kraja *st* funkcija neće ukloniti ni jedan znak. Ukoliko se izborom velikog *i2* zahteva brisanje slova koja na postoje funkcija će vratiti segment stringa *st* od počtka do znaka *(i1 1)*–Primer s. = *Pascal*, *delete (s 3 2)* daje vrednost Psal

procedure insert (st1, st2, i)

Umetanje stringa *st1* u string *st2* i to počevši od pozicije *i* Ukoliko je rezultujući string duži od 255 znakova, višak će biti odsečen. Primer e = *'Pascal'*, *insert (' i e, 4)* daje 'Pes t cal

function length (st) i

Vraća dužinu stringa *st* Ako je string prazan dužina je nula. Primer e = *'Pascal li length (a)* daje vrednost 6.

function pos (st1 st2) iby

Pronalazi prvo pojavljivanje stringa *st1* u stringu *st2* i vraća njegovu poziciju ili nulu ako *st1* nije pronađen, Primer e = *'Pascal'*, *b =pos ('sc', a)* daje vrednost 3.

procedure str (ir, st)

Konvertuje bilo koji ceo ili racionalan broj *ir* u string *st* Konverzijom se može upravljati kao i u naredbi *write str (25 4 a)* daje string ' 25' Konverzijom racionalnih brojeva se upravlja se *iznaz sim-na-decimale*

procedure val (st, ir, i)

Konverzija stringa *st* u ceo ili racionalan broj *ir* Ceo broj *i* opisuje uspeh konverzije *i =0* označava da je konverzija uspela dok će, eko konverzija ne uspe, *i* vratiti poziciju prvog znaka koji je pravio pro bleme. Vredi pomenuti da se funkcija *val* na može, kao u nekim bejzicima, koristiti za izračunavanje vrednosti kompletnog izraza koji je zadat kao string mogu se konvertovati jedino konstante zadate u skladu se sintaksom Turbo Pascala.

9. Slogovi i datoteke

Iako paskal primamo nije namenjen redu se bazama podataka, ISO standardima su uvedeni veoma kompletni slogovi i relativno nepotpune datoteke kojima se može pristupati isključivo sekvencijalno Borlandovi programeri su uveli brojne ekstenzije koje obezbeđuju ne samo slučajan pristup neophodan za red se bazama podataka već i pisanje sistera skih programa koji operišu sa datotekama na disku i periferijskim uređajima koji što se paskala tiče predstavljaju najobičnije datoteke.

Počnimo od slogova na slici 9 1 vidimo dekla racije koje bi bile ugrađene u neki paskal program koji bi se bavio kadrovskom evidencijom svaki službenik ima slog u kome je dalm imena, prezimena

```
const  ime_duz  = 10;
       prez_duz = 25,
       max_zap  = 100,
type
       radnik = record
                  ime      string [ime_duz],
                  prezime  string [prez_duz];
                  pol      (muskarac, zena),
                  staz     0 40,
                  sati     0 .300;
                  plata    real
                end;
var    zaposleni  array [1 max_zap] of radnik,
       uneseni    radnik,
```

Slika 9.1

pola I radnog staža upisan broj radnih sati I ukupna plata.

Tip *radnik* je definisan kao *record* (slog) koji se sastoji od polje *ime prezime, pol staž sati* I plata svako od ovih polja je deklarisano na odgovarajući način što znači da u svakom zapisu imamo dva string ga jednu promenljivu specijalnog tipa dva cela I jedan racionalan broj Definicija sloga se kao i ave druge struktura u paskalu završava se *end* dok su polja razdvojena separatorom ;

Slogovi ne moraju da budu vezani se spoljnom memorijom u primeru 9 1 uvedeni je niz *zaposleni* koji se sastoji od najviše 100 slogova *radnik* koristili smo dakle, činjenicu da elementi matrica mogu da budu na samo sleiani već i komplikovana strukture Što se operacija se slogovima tiče možeme da prenesemo sadržaj jednog sloga u drugi (npr *zaposleni[i] =uneseni*) ali nam je daleko važnije mogućnost da pojedinačno pristupamo nekim poljima. Kao što za pristupanje elementima matrica koristimo srednje sagrade tako za pristup pojedinim poljima koristimo tačku ako je treći saposleni muškarac, napisaćemo nešto poput

zaposleni[3] pol = muskarac

Stvar na postaje malo komplikovanija ako definišemo slogove čiji su elementi drugi slogovi

U primeru 9 2 tip *radnik* je proširen dodavanjem datuma rođenje taj je datum novi slog koji se sastoji od dana, meseca I godine Na istoj slici vidimo kako se unosi podatak de je treći radnik *roden* 12 marta 1952 godine *zaposleni[3] nse* upozorava da se radi o trećem zaposlenom *roden* označava da rečmo se poljem *roden* dok *dan*, *mesec* I *godine* označa-

Pascalu omogućavaju korišćenje usluga MS DOS-a odnosno PC DOS-a sastavni deo ovog modula su, na primer, procedura *GetTime*, *GetDate*, *DiskSize* i *Kd*. Definisani tipovi i konstante ovog modula obezbeđuju pisanje interaptskih rutina i drugih mašinsko zavisnih procedura na Turbo Pascalu.

Crt unit obezbeđuje opcije za očitavanje tastature i ispis teksta na ekranu i generisanje zvuka koje su precizinije od standardnih procedure *read* i *write*. Modul se međutim, ne bavi grafikom.

Printer unit je sasvim kratak i obezbeđuje komunikaciju sa štampačem (slika 7.4). Ovaj modul koristi **Crt**.

Graph unit je zapravo moćan skup grafičkih rutina koje su kompatibilne sa CGA, EGA, Herkules, AT&T 400 MCGA, 3270 PC i VGA standardima. Ovaj modul koriste **Crt** i ne mora biti integralni deo datoteke *turbo.tpu* smešten je u *graph.tpu*

Graph3 unit obezbeđuje kompatibilnost sa grafičkim programima pisanim za verziju 3 Turbo Pascala. Smešten je u datoteku *graph3.tpu* što znači da nije element *turbo.tpu*

Turbo3 unit obezbeđuje kompatibilnost sa programima pisanim za verziju 3 Turbo Pascala. Najznačajnije komponente ovog modula su predefinisane pseudo datoteke *Kbd* logička promenljiva *CBreak* i varzije *MemAvail* (*MaxAvail*) koje vraća broj slobodnih paragrafa a ne broj slobodnih bajtova. Modul je smešten u datoteku *turbo3.tpu* što znači da nije element *turbo.tpu*

```
program prt (input, output),
uses printer;
var a, b  integer;
begin
   a := 3 ; b := 4,
   writeln (lst, 'Zbir brojeva ',a,' i ',b, je'),
   writeln (lst  a + b)
end
```

Slika 7 4

8. Stringovi

Sa mnogo prava se može tvrditi da je rad sa alfanumerićima Ahilova peta standardnog paskala. Jezik u kome svako petljenje sa interaktivno unesenim tekstom zahteva "kilometarski" kod prepun raznih programerskih trikova teško može da se progiesi pogodnim za praktično programiranje! Zato su razni autori raznih paskal kompajlera predvideli razne (i na žalost, međusobno slabo kompatibilne) ekstenzije. Borlenodova ekstenzije predstavljaju dobro zaokruženu celinu koje je nerocito pogodne za poznavaoce bejzika, koji će se osećati "kao kod kuća"

Alfanumerici se kao što znamo deklarišu sa *string[n]* gde je *n* maksimalne dužina stringa koja ne sme prelaziti 255 znakova. eko se *n* izostavi podrazumeva se 255. Učitavanje i štampanje alfanumeričkih promenljivih se ne razlikuje od učitavanje i štampanje promenljivih bilo kog skalarnog tipa. možemo da napišemo *readln(tekst)* pa će promenljivoj *tekst* biti dodeljen skup znakova unesenih preko tastature. U sam string spada i marker kraja što znači da ne moramo brinuti o dužini alfanumerika niti uvodili posebne promenljive računar uvek zna koji je deo maksimalnog deklarisanog prostora zauzet!

Stringovi sa mogu nadovezivati (stručni izraz je *konkatenacija*) i to primenom običnog pluso program 8.1 će nadovezati reč Pascal na reč Turbo i tako kreirati string 'Turbo Pascal

```
program konkat (input, output),
var a, b, c: string,
begin
   a := 'Turbo ',
   b := 'Pascal',
   writeln (a + b)
end
```

Slika 8.1

Dopušteno je i korišćenje relacionih operatora = <>, <, >, <= i >= pomoću kojih možemo da ustanovimo da li su stringovi jednaki ili je neki od njih "veći" od drugog pri poredenju se kao i u bejziku uzima u obzir isključivo ASCII tabela što znači da je tekst Pascal "manji" od teksta pascal i to lekju živo zbog razlike između velikih i malih slova.

Ostatak ovog poglavlja posvećujemo procedurama i funkcijama koje operišu sa stringovima. Za opis argumenata koristili smo sledeću konvenciju

by	byte
wo	word
si	shortint
li	longint
i	integer
ll	bilo koji od tipova *by, wo si li i*
r	real
lr	bilo koji od tipova *li r*
bo	Boolean
ch	char
st	string

function concat (st1, st2, ... , stn) st

Konkatenacija odnosno nadovezivanje stringova *st1 st2, ... stn* i vraćanje rezultujućeg stringa. Ova funkcije je u izvesnom smislu jedinstvena pošto ima promenljiv broj argumenata

function copy (st1, i1, l2) st

Izdvajanje podstringa koji ima *l2* znakova počevši od znaka čiji je redni broj u stringu *st1 i1*. Ukoliko *i1* pokazuje iza kraja *st1* funkcije vraća prazan string. Ukoliko se izborom vellkog *l2* zahteva kopiranje slova koja ne postoje funkcija će vratiti

koju ćemo opisati u daljem tekstu - ako je opis prekidača {$B } podrazumevana vrednost je minus (isključeno) pri čemu je programeru pružena mogućnost da sa {$B + } aktivira prekidač

{$B-} - izračunavanje Bulovih izraza

Kada je ova opcije aktivna svaki logički izraz se garantovano izračunava od početka do kraja. Zar može drugačije? Zašto da ne... eko je vrednost promenljive e pat, izraz *(e –3) and (b = 4)* je fatas bez obzira ne vrednost promenljive *b* što znači da računanje vrednosti drugog dela izraza predstavlje čisa gubitak vremena. Programar ća međutim ponekad požeiati da forsira ovo izračunavanja kako bi sa pot vrdilo da je vrednost *b* definisana u trenutku izvršavanje ove naredbe. Vredi dodati da su standardi ISO Pascala po ovom pitanju prilično nedefinisani tvrdi se jedino da *se standardom na može garantovati da će logički izraz uvek biti računat do kraja*. Turbo Pascal to po potrebi garantuje ili ne garantuje

{D + } - generisanje linijskih brojeva

Turbo Pascal saradnje sa raznim debagerima (Borland preporučuje dobro poznati program "periskop") tako što u kod ugraduje linijske brojeve pojedinih procedura na osnovu kojih se u potrebnom momentu može utvrditi kojoj liniji izvornog programa odgovara neki segment objektnog koda. Uklanjem ovih linije kod se skraćuje i (naznatno) ubrzava ali je saradnja sa debagerom nemoguća. Turbo Pascal čak neće umeti ni da locira liniju u kojoj je nastupila greška!

{$F-} - forsiranje dugih skokova

Ukoliko programar u izvorni program ugradi opciju {$F + } poziva svih procedura biće prevedeni kao dugi skokovi.. ukoliko je ova opcija neaktivna, Turbo Pascal će generisati dugi skok samo eko je ovaj zaista neophodan. Korišćenje opcije F ima smisla jedino ako povezujemo paskal programa sa rutinama pisanim ne nekim drugim jezicima.

{$I + } - kontrola ulazne/izlaznih greški

Ukoliko pri izvršenju nekog programa dođe do ulazno/izlazne greške (nepostojeća datoteka nedozvoljen pristup prepun disk) izvršavanje programa se obustavlja i računar prijavljuje grešku. Ponekad je međutim pogodno da se izvršavanje nastavi i da greška prijavi sam program. u tom slučaju u kod treba ugraditi {$I } a zatim koristiti funkciju *Ioresult* ne način koji ćemo opisati u jednom od sledećih poglavlja. Ova je po svoj prilici najčešće korišćena prekidačka direktiva.

{$L + } - kontrola linkera

Interni linker ugraden u Turbo Pascal je dvoprolazni pri čemu se međurezultati mogu držati u memoriji ili na disku {$L + } označava da su baferi u memoriji što ubrzava rad {$L } ćemo koristiti ako naš računer nije snabdeven dovoljnim RAM-om

{$N-} - priseuzve ertimetičkog koprecesora

Ukoliko je opcije N u svom podrazumevanom stanju tj. isključena, Turbo Pascal generiše kod za rad sa racionalnim brojevima. {$N + } se druge strane znači da se kompletna računica prepušta aritmetičkom koprocesoru

{$R-} - kontrola opsega

Pre izvršavanja naredbi poput *a[i] :=0* računar treba da provari da li se vrednost promenljive *i* nalazi u granicama koje su odredene kada je niz deklarisan. Ova provera zahteva ugradnju značajnog broja instrukcija koje produžavaju i, što je možda još važnije usporavaju prevedeni program pa ća je Turbo Pascal jednostavno preskočiti. U toku razvoja programa je međutim, pogodno koristiti {$R + } i tako osigurati prijavljivanje greške u svim konfliktnim situacijama.

{$S + } - provera prekoračenja steka

Ukoliko je program veoma složen predvideni prostor ne steku mu ne mora biti dovoljan što znači da ća se neke povratne edrese jednostavno izgubiti, odgovarajuća mašinska instrukcija RET će tada po prilici stvari izazvati krah sistema. Kontrolu prekoračenja steka, dakle, smemo da isključimo samo eko je program detaljno testiran.

{$T-} generisaoje TMP datoteka

Prilikom prevodenja programa čije je odredište disk možemo da zahtevamo i generisanje odgovarajućeg TMP fajla koji će biti neophodan nekim debagerima. Sastavni deo paketa Turbo Pascal 4.0 je i usiužni program TPMAP.EXE koji TMP datoteku pretvara u MAP i tako je prilagodava simboličkim debagerima kao što je Microsoftov *CodeView*. Korišćenja opcije {$T + } je besmisleno ako se prevedeni program smešta u memoriju i eko je korišćena opcija {$D }

{$V + } - kontrola stringova

Prilikom prenošenja parametara u procedure Turbo Pascal zahteva da formalni i stvarni argument budu identično deklarisani što pored ostalog znači da stringovi moraju biti iste dužine. Korišćenjem opcije {$V } omogućavamo prenošenje stringova bez provere dužine što je prilično pogodno ati i neusaglašeno se koncepcijom paskala kao jezika.

Parametareke direktive

Parametareke direktive u skladu sa svojim imenom obezbeđuju neke operacije kojima je neophodan parametar u vidu imena neke datoteke ili kapacitata memorije.

{$I ime} - umetanje paskal programa

Direktiva *Include* obezbeđuje umetanje kompletnog programa iz datoteke *ime* u tekući program baš kao da je segment upisan red po red. Turbo Pascal podrazumeva da je program upisan u datoteku čije je ekstenzije PAS i koje se nalazi u direktorijumu specifikranom u okviru *Options/Directories/Include* ili u radnom katalogu. U ovoj datoteci mogu da se nadu druge *Include* direktive i tako dalje ograničenje je jedino limit Turbo Pascala

koji u istom trenutku ne može otvoriti više od osam datoteka.

Vredi pomenuti i jedno ograničenje direktive *Include* - dopušteno je nmetenje kompletnih naredbi ali na i njihovih segmenata!

{$L ime} - umetanje prevedenog programa

Direktiva *Link* obezbeđuje povezivanje programa pisanih na Turbo Pascalu sa programima pisanim na drugim programskim jezicima kao što su fortran C ili asembler. Standardna ekstenzija datoteke je OBJ a standardni direktorijum u kome se ona traži može da se specificira sa *Options/Directories/Object*. Povezivanje Turbo Pascala sa drugim jezicima, na žalost izlazi iz domena ovog uvoda.

{$M stack heapmin,heapmax} - alokacija memorije

Uz pomoć ove opcije definiše se maksimalni memorijski prostor alociran za neke potrebe Turbo Pascala. *Stack* označava prostor alociran za stek i kreće se između 1024 i 65520 bajta. *Heap* je struktura na kojn se smeštaju vrednosti promenljivih i mnogi drugi parametri i kreće sa između *heapmin* i *heapmax* prvi broj mora da bude u intervalu [0 655360] a drugi u intervalu [heapmin 655360]

{$U ime} - Osteteka sa modulima

Turbo Pascal obezbeđuje kreiranje posebnih celina koje se zovu moduli (*units*). Podrazumeva se da je modul koji se zova DEMO smešten n datoteku DEMO.TPU all se ova konvencije može izmeniti ako je DEMO smešten n $VE TPU koriščenio *uses $u {sve} demo*

Kondiciona direktive

Normalno bi bilo očekivati da se prevođenjem lekog izvornog programa uvek dobija lsti objektni kod. Ne mora, međutim, uvek biti baš tako u fazi testiranje programer će poželeti da ugradi iteka segmente koji gramda nebitni se stanoviška izvršavanja, olakšavaju kcriranje greški ovakve segmente svakako izbacujemo iz finalnog programa all se nikada ne može garantovati da se nećemo vratiti testiranju! Zbog svega toga Turbo Pascal 4 0 obezbeđuje nslovnu kompilaciju neki segmenti koda se prevode samo ako su se stekli određeni uslovi

Opšti oblik kondicione direktive je {$Fxxx} .. {$ELSE} {$ENDIF} gde su trima tačkama označeni segmenti paskal programa. Ukoliko je uslov xxx ispunjen prevodi se segment izmedu {$Fxxx} i {$ELSE} odnosno ako {$ELSE} segment ne postoli {$Fxxx} i {$ENDIF} nkolko uslov xxx nije ispunjen prevodi se segment izmedu {$ELSE} i {$ENDIF} odnosno ništa ako {$ELSE} sekcije na postoji

Uslov xxx može da bude *DEF ime NDEF ime* i *OPT prekidač DEF ime* je *true* ako je simbol *ime* definisan, *NDEF ime* je true ako simbol *ime* nije definisan tlok je *OPT prekidač* (npr {$IFOPT R+ }) *true* ako je prekidač u odgovarajućem stanju (npr R aktivan) Simbole definišemo direktivama {$DEFINE ime} e njihova definicije poništavamo sa {$UNDEF ime}

Turbo Pascal podržava neke unaprad definisane simbole koji obezbednjn obaveštenje o konfiguraciji računara na kome se program prevodi. Tako je simboi VER40 uvek definisan i označava da se program pravodi na verzij 4 0 Borlandovog kompajlera MSDOS je uvek definisan označavajući da se prevođenje vrši na nekoj MS DOS ili PC DOS mašini CPU86 je definisan kako bi se naglasio da mikroprocesor pripada Intelovoj familiji 80x86 dok je CPU87 definisan samo ako je aritmetički koprocesor prisutan Na početak svakog programa, dakle ima emisla ugraditi red *{$IFDEF CPU87} {$N+} {$ELSE} {$N-} {$ENDIF}* i tako obezbediti da se se racionalnim brojevima operiše softverski samo ako je to stvarno neophodno.

Tretiranje greški

Niko nije savršen pa čak ni programer koji koristi Turbo Pascal u toku razvoja nekog programa svakako će se pojaviti značajna količina sintaksnih izvršnih i logičkih greški koje treba pronaći i otkloniti. Ponašanje Turbo Pascala u konfliktnim situacijama je veoma logično što znači de lepravljenje greški ne bi trebalo da bude preteško.

Ukolko se pri prevođenju programa nalde na grešku kompajler ustupa mesto editoru kurzor je da bi stvar bila još jednostavnija pozicioniran na problematičnu naredbu dok statusna linija opisuje grešku sve je spremno za ispravku i novi pritisak na Alt R Ne može se jasno garantovati da se greška nalazi u liniji u kojoj je detektovana ako je neka promenljiva nedefinisana moraćete da potražite kurzor do početka programa ili do početka procednre i da tamo dodate odgovarajući red Većina greški se međutim može otkloniti na licu mesta.

Greška nastala pri izvršavanjn programa manifestuje se na identičan način pri čemn je verovatnoća da je kurzor stvarno na pogrešnoj naredbi nešto manja varovatno ćete morati da pretražujete okolni segment programa ili čak neke udaljene procedure

Vredi pomenuti i ponašanje sistema pri radn sa modulima. Ukolko je greška nastupila u TPU datoteci e na n glavnom programu računar će automatski učitati odgovarajući modul i pozicionirati kurzor u problematični red Posle se ovakvim automatskim učitavanjem nepovratno gubi program upisan u memoriju Turbo Pascal će vam pružiti priliku da ga snimite ako pritisnete Esc, modul neće biti ni učitan što je prilično zgodno kada znate da greška nije u modulu nego n glavnom programu koji pripremu pogrešne podatke!


```
UNIT stat,

INTERFACE

procedure clear;
procedure newxy   (x,y  real),
procedure findmbr (var m, b, r: real),

IMPLEMENTATION

type struct  =      record
                       nn, sx, sy,
                       sx2, sy2, sxy: real,
                    end,

var   statblock:  struct,

procedure clear;
begin
   with statblock do
   begin
      nn := 0;  sx := 0;  sy := 0,
      sx2 := 0, sy2 := 0, sxy := 0
   end
end,

procedure newxy (x, y  real),
begin
   with statblock do
   begin
      nn := nn + 1; sx := sx + x; sy := sy + y;
      sx2 := sx2 + x*x, sy2 := sy2 + y*y,
      sxy := sxy + x*y
   end
end,

procedure findmbr (var m, b, r: real),
var temp: real,
begin
   with statblock do
   begin
      temp := nn*sx2-sx*sx,
      m := (nn*sxy-sx*sy)/temp,
      b := (sx2*sy - sxy*sx)/temp,
      r := (nn*sxy-sx*sy)/
           sqrt(temp*(nn*sy2-sy*sy)),
      r := abs(r)
   end
end,

begin
   clear;
end
```

Slika 7.2

Primer 7.3 ilustruje upotrebu modula *stat* sa slike 7 2 računar učitava seriju parova (X Y) e onda poviači najbolju pravu kroz njih i određuje koeficijent korelacije

Rad sa modulima zahteva malo dodatne brige oko prevođenja I dalje možemo da pritisnemo Alt R i tako pravademo glavni program u koji će potpuno automatski biti uključeni i svi moduli all se prethodno treba pobrinuti da ti moduli zaista budu prevedeni - ako smo zaboravili da prevedemo modul izmene

```
program statistika (input, output),

uses stat;

var   n, i: integer;
      x, y, m, b, r: real;

begin
   write ('Koliko će parova biti? '),
   readln (n),
   clear;
   for i := 1 to n do
   begin
      write ('x[',i,'] = '),
      readln (x),
      write ('y[',i,'] = '),
      readln (y),
      newxy (x, y),
   end,
   findmbr (m, b, r),
   writeln ( Koeficijent pravca      ', m:10.5),
   writeln ('Odsečak                 ', b:10 5),
   writeln ('Koef. korelacije        ', r:10.5)
end
```

Slika 7 3

koje smo unell se jednostavno na uzimaju u obzir Zato je pogodno koristiti opcije *Make* i *Build* iz **Compile** menija i to na način opisan n okviru trećeg poglavlje ovog uvoda

Ograničenja mikroprocesora 80x86/88 učinila su da jedan segment ne može prevazići 64 kilobajta što važi i za Tnrbo Pascal Moduli međutim pred stavljaju elegantan način da se ovo ograničenje prevazide ograničenje od 64 kilobajta važi za svaki modul posebno što znači de kod više modula uzet zajedno može itekako prevazići 64 kilobajta. limit je ukupan RAM vašeg kompjutera umanjen za određeni segment koji zauzimaju DOS i razidentni programi.

Sistemski moduli

Sećate li se datoteke *turbo tpu* koja se mora nalaziti na radnoj disketi ili u radnom direktorijumu? Radi se o modulin koji sadrži veoma važne deklaracl je funkcije I procedure bez kojih Turbo Pascal uopšte na bi funkcionisao takozvana *izvršna biblioteka* Turbo Pascala zove se *system unit* i nalazi se u datoteci *turbo tpu* Obzirom da su rutina iz sistem skog modula stalno neophodna sintaksom nije pre dviđena obavaza da se svaki program započne sa *uses system* Datoteka *turbo tpu* međutim sadrži i druge važna module koje ćemo ovde samo nabrojati dok ćemo opširnlji opis ostaviti za finalno poglavlje Svaki od modula se nkolko ga koristimo mora eksplicitno navesti iza *uses* što je prilično zgodno modul ćemo referencirati samo ako zaista namera vamo da ga koristimo što spračava da EXE datoteka bude pretrpana kodom koji se uopšte ne poziva!

DOS unit sadrži razna deklaracije funkcije I procedure koje programima razvijanim na Turbo

7. Rad sa modulima

Moduli (ili u terminologiji Turbo Pascala, *units*) predstavljaju značajno pobojšanja verzije 4.0 koja je prilagođila Turbo Pascal PC-jevom mikroprocesoru i operativnom sistemu i olakšalo programiranje odnosno ispravljanje greški.

Iako nisu predviđeni standardima paskala, moduli su sasvim u skladu sa njegovim duhom strukturirano programiranje zahteva de se funkcionalne jedinice izdvoje u posebne funkcije ili procedure ali ne standardnom paskalu te funkcije i procedure moraju biti uplsane u istu datoteku u kojoj je i glavni program. Ukoliko se dakle ukaže potreba za izmenom u nekoj od procedura, računar nema izbora nego da prevede kompletan program zajedno sa svim ispravnim procedurama i funkcijama. Zar ne bi bilo prirodnije da se svaka funkcionalno povezana grupa procedura odvoji u posebnu datoteku da se nezavisno testira, prevede i kada je god to potrebno, uključi u finalni program? Takva funkcionalno povezana grupa deklaracije procedura i funkcija sa ne Turbo Pascalu 4.0 zove modul (*unit*)

Struktura modula

```
unit ime;
interface
uses lista,
{ javne deklaracije }
implementation
{ privatne deklaracije }
{ procedure i funkcije }
begin
  { inicijalizacioni kod }
end
```

Slika 7.1

Struktura modula prikazana na slici 7.1 je u mnogo čemu slična strukturi samog programa. *Unit ime* je da početnomo od početka, (ovoga puta neophodni) skrivalaret deklaracije *program* računaru se stavlja do znanja da je ono što sledi modul e na racimo, glavni program e onda se tom modulu daje ima. Sledi interfejs sekcija preko koje modul komunicira sa nadređenim modulima svi tipovi konstante, promenljive funkcije i procedure deklarisane u ovoj sekciji su "vidljiva" za pozivajući modul interfejs sekcije započinje službenom reči *interface* i savršava se službenom reči *implementation* zanimljivo je da iza ova dva reči ne treba stavljati separator , što ne neki način nije u skladu se koncepcijom paskala.

Prva (neobavezna) linija interfejs sekcije je *uses* uz njenu pomoć modul može da koristi druge module navedena u listi. Ostatak je sasvim sličan početku paskal programa sekcija tipova, promenljivih konstanti i na kraju lista procedura i funkcija. Kažemo *lista* procedura i funkcija jer se u interfejs sekciji navode samo imena i argumenti funkcija i procedura dok se njihov radni deo "skriva" u implementacionu sekciju kleuzula *forward* ovde nije potrebna niti dopuštena.

Implementaciona sekcija predstavlje privatni deo modula ovde se ponavlja deklaracija svake procedura propraćena njenim radnim delom pisanim u skladu se simlakom Turbo Pascala ukoliko se za tim ukaže potreba ovde se može koristiti kleuzula *forward* što znači da se deklaracija neke funkcije ili procedure u okviru kompletnog modula može javiti čak i tri puta Implementaciona sekcija ne mora da sadrži samo redne delove procedura deklarisanih u okviru interfejs sekcije mogu se deklarisati novi tipovi konstante promenljiva procedura i funkcije koje su nevidljive za pozivajući program i koje "javne" procedure koriste da bi obavile svoj posao Implementaciona sekcija sva u svemu, predstavlja pravi paskal *program* koji je dopunjen (dodate neobaveznim) glavnim programom koji počinje se *begin* e završava se *end* (ne zaboravite tačku!) Ovaj pseudo glavni program se automatski izvršava pre pravog glavnog programa i treba da obuhvati naredbe koje će izvršiti inicijalizaciju promenljivih koje su neophodne za normalan rad modula. Nema nikakve prepreke da u okviru ove inicijalizacije postavimo i neke privatne vrednosti deklarisane u okviru implementacione sekcije modula one će dođuše, bal nevidljive za nadređeni program ali će nas vrednosti "začekati" kada god modul bude pozvan.

Primer 7.2 predstavlja ilustraciju svega o čemu smo do sada govorili napisali smo statistički modul koji se ugrađuje u svaki program kome je potrebna takozvana linearna regresija odnosno povlačenje prave linije koja u minimalnoj mogućoj meri odstupa od serije eksperimentalno određenih tačaka. Modul treba smestiti u datoteku *stat.pas* e onda njegovim prevođenjem (prevođenje se Alt F) je ovde nemoguće jer se modul ne može izvršiti, treba odabrad **Compile** meni promenlti odredlite (*Destination disk*) i najzad izvršiti *Compile*) i kreirati datoteku *stat.tpu* (TPU je standardna ekstenzija prevedenih modula) koju će glavni program koristiti na način koji ćemo za trenutak upoznati

Upotreba modula

Moduli se koriste sasvim jednostavno na samom početku pozivajućeg modula (kažemo *pozivajućeg modula* e na *glavnog* programa jer i moduli mogu da koriste druge module) treba napisati *uses lista* gde je *lista* spisak korišćenih modula razdvojenih zarezima U izvršnom delu modula mogu se slobodno koristiti definicije, funkcije i procedure pomenute u interfejs sekciji pedređenog modula bez potrebe da se ona na bilo koji način deklariču Zanimljiva se situacija javlja jedino ako se u po njeranihjl starijem modulu deklariše neka promenljiva čije je ime jednako promenljivoj iz modula Prednost toda ima glavni program ali je ipak omogućen I pristup promenljivoj iz modula I to sa ima *module ime_promenljive*

6. Osnove Turbo Pascala

Iako smo u uvodnom poglaviju naglasili da je ovaj umetak namanjen poznavaocima paskala koji bi želeli da iskoriste sve mogućnosti Borlandove varijante popularnog programskog jezika ovo poglavlje posvećujemo nekim osnovnim osobinama paskala I brojnim razlikama između Borlandovih I ISO standarda počstnici će na taj način stati utisak o stvarima koje treba da upotrebljavaju dok će za iskusnije programare tekst biti koristan repetitorijum

Kostur programe

Počšećemo od jednostavnog paskal programa sa slike 6.1 koji pronalazi aritmetičku sredinu neko iiko unetih brojeva.

Neka bitne osobine se odmah uočavaju paskal programi se pišu malim slovima pri čemu bi možda bilo tačnije reći da Turbo Pescal jednostavno ignoriše razlike izmedu velikih i malih slove programe pišemo onako kako nam odgovara all je korišćenje (za gledanje mnogo prijatnijih) malih slova veoma ubičajeno Treba dakle zapamtiti da paskal ni po čemu ne razlikuje promenljive koje smo nazvali *JEDAN, jedan* i *JeDaN*

Svakako ste primetili da sva naredba se slike 6 i na početnjo od početka reda. kažemo da je program "nazubljen" Nazubljivanje u stvari nije neophodno. eve bi bilo u redu da smo naredba pisali od prva pozicije u redu ali se s pravom smatra da uvlačenje pojedinih redova doprinosi čitljivosti programa utoliko što vizualno izdvaja njegove delove Uobičajeno je da se u okviru svakog bloka naredba uvlača za tri pet blanko simbola all se ovaj broj ponekad smanjuje u cilju uštede prostora.

Standardni separator instrukcije u paskalu je simbol , ovaj znak morate stavljati izmedu dva instrukcije čak i kada se one nalaze u različitim redovima U prvim danima navikavanje na paskal redovno ćete izostavljati neophodni separator što će rezultirati različitim porukama o greści koje nije uvek lako identifikovati! Možete se ipak utešiti time da je uvođenjem separatora omogućeno da se jedno naredba prostire u više redove što smo u raznim paskal programima koje smo objavljivali u "Računarima obično koristili skraćujući "gigantske" linije štedimo dosta prostora

Svaki program počinje odredenim brojem neophodnih deklaracija na paskalu morate de špišete gomilu redova pre nego što nešto stvarno počne da se dogada Prva linija *(program prosek (input,output))* govori računaru da je ono što sledi program e na liste baleuluka odreduje ima programa (*prosek*) i tvrdi da će program učitavati nekakve podatke (*input*) i ispisivati neke rezultate (*output*) Vradi reči da je ovaj red sa aspekta Turbo Pascala sasvim nepotraban I da će program savršeno raditi I kada ga izostavite večina programera ipak započinje program ovom deklaracijom uvažavajući na taj način standarde "pravog" paskala

Druge linija našeg programa je komentar u paskalu sa komentari okružuju vitičastim zagradama što znači da se komenter može prostirati u više redova. Obzirom da su prve verzije paskala imple mentirane na računarima sa skromnim setom znakova standardima je predviđeno da se početak komentara ravnopravno može obeležiti se (* e kraj se *). Konstruktori Turbo Pascale su zgodno iskoristili ovu alternativu ponekad ćete poželeti da neki deo programa privremeno "isključite iz opticaja" e da ga ipak ne uklonite iz datoteke Na raspolaganju je naravno ueiovno prevođenje opisano u prethodnom poglaviju all će većina programera repotrebni segment jednostavno pretvoriti u komentar Posebni problemi se, začudo neće javiti čak ni ako u izdvojenom segmentu već postoje neki komentari linije koje uklanjamo treba obeležiti jednom (npr (* *)) e prave komentare drugom ({ }) varijantom

Sledi lista deklaracija koja je započete službenom reči u *var* (od *variable* promenljiva) svake promenljiva koju ćemo koristiti mora da bude deklarisana na samom početku programa Zatim se navode sve funkcije i procedure I na kraju glavni program koji počinje se *begin* e završava se se *end* tačka iza *end* je obavezna kako na standardnom paskalu tako I na Turbo Pascalu jer kompajleru traba dati do znanja da je radni deo programa završen I da se eventualni ostatak datoteke slobodno može ignorisati.

```
program prosek (input,output);

{
bračunava aritmetičku sredinu
N zadatih brojeva
}

var  i,n   integer;
     e,e,  real;


begin
   write ('Koliko će brojeva biti? ');
   readln (n),
   s.=0;
   for i =1 to n do
   begin
      readln (e);
      e =e+e
   end,
   writeln ('Prosek je= ', s/n)
end.
```

Slika 6.1

Promenljive i tipovi

Videli smo de sve promenljive u paskalu moraju da se deklarišu pri čemu deklaracija obuhvata ime I tim. Što se imena tiče stvar je sasvim jednostavna poštuješ bilo koji niz od najviše 63 slova I cifre čiji je prvi simbol slovo za razliku od standardnog Turbo Pascal dopušta I korišćenje znaka _ koji može da učini ime promenljive čitljivijim. Slika 6.2 prikazuje

```
absolute  file            mod        shl
and       for             nil        shr
array     forward         not        to
begin     function        of         type
case      goto            or         unit
const     if              packed     until
div       implementation  procedure  uses
do        in              program    var
downto    inline          record     while
else      interface       repeat     with
end       interrupt       string     xor
external  label           set
```

Slika 6.2

rezervisane reči Turbo Pascala tj. imena naredbi koje se ne smeju koristiti kao imena promenljivih

Što se tipova tiče, najčešće ćete koristiti skalarne promenljive koje mogu da budu celobrojne, racionalne, logičke ili alfanumeričke. Celobrojne promenljive se deklarišu korišćenjem službene reči *byte* ([0..255]), *shortint* ([-128..127]), *word* ([0..65535]), *integer* ([-32768..32767]), i *longint* ([-2147483648..2147483647]) što očito predstavlja značajno poboljšanje u odnosu na standardni paskal. Racionalne promenljive su takođe podeljene na grupe

Tip	Opseg	Tačnost
single	$[1.5*10^{-45}..3.4*10^{38}]$	7-8
double	$[5.0*10^{-324}..1.7*10^{308}]$	15-16
real	$[2.9*10^{-39}..1.7*10^{38}]$	11-12
extended	$[1.9*10^{-4951}..1.1*10^{4932}]$	19-20

Ukoliko se, međutim, niste opremili aritmetičkim koprocesorom, moći ćete da koristite isključivo tip *real* baš kao i u standardnom paskalu.

Logičke promenljive se, u čast Džordža Bula (George Boole, 1815-1864) zovu Bulova i deklarišu se *Boolean* iako je kompajleru sasvim svejedno uobičajeno je da ova reč započne velikim slovom. Logička promenljiva može da ima samo vrednost *true* ili *false* pri čemu je standardima paskala utvrđeno da je *false*<*true* i da je *Ord(false)*=0 i *Ord(true)*=1

Alfanumerička promenljiva se deklariše sa *char* i "pamti" samo jedan ASCII znak. Ovo očito predstavlja veliki korak nazad u odnosu na *fortran* 77 ili bejzik pa su konstruktori Turbo Pascala uveli i tip *string* koji se tretira na poznavaocu bejzika sasvim prirodan način: promenljivoj dodeljujemo proizvoljni string čija dužina ne sme da pređe 255 znakova a zatim taj string slobodno analiziramo, štampamo i obrađujemo na način koji će biti opisan u jednom od sledećih poglavlja. String promenljiva se deklarišu sa *string[n]* gde je *n* neoznačen broj koji definiše maksimalnu dužinu stringa, ako se izostavi podrazumeva se 255.

Jedna od zanimljivih osobina paskala je da niste ograničeni samo na osnovne tipove podataka: možete da uvodite i sopstvene tipove što će pokazati i primer 6.3

```
program test (input,output);

type pozitivan = 1..32767;
     cifra     = 1..9;
     jezici    = (fortran, cobol, algol, bejzik,
                  pascal, modula2);
     boje      = (crna, zuta, zelena, bela);
     kompjuteri = (Spectrum, Commodore,
                  Amstrad, Atari,
                  BBC, IBM_PC);

var  brojac:   jezici;
     ij:       pozitivan;
     firma:    kompjuteri;
     tapet:    boje;
```

Slika 6.3

Ispred sekcije promenljivih (*var*) ubacili smo novu sekciju koja je započela sa *type*: radi se o deklaracijama tipova. Pogledajmo najpre tip *pozitivan* koga smo opisali kao *1..32767*: dve sukcesivne tačke označavaju da promenljiva tipa *pozitivan* može da ima bilo koju celobrojnu vrednost između 1 i najvećeg celog broja koji računar poznaje. Kakav je smisao uvođenja ovoga tipa kada smo jednostavno mogli da koristimo tip *integer*? Ukoliko neka veličina po svojoj prirodi mora da bude pozitivan broj njen zalazak u "negativne vode" označava da se program našao u nepredviđenoj situaciji koju je najbolje signalizirati: korisnik će tada znati da su rezultati verovatno pogrešni. Paskal prevodilac će, dakle, ugraditi u rezultujući mašinski program naredbe koje proveravaju da li se vrednosti promenljivih nalaze u predviđenim limitima i ispisuju odgovarajuće poruke u konfliktnim situacijama, ove će nam poruke pomoći pri testiranju programa i uklanjanju bagova. Nema, jasno, potrebe da se ograničimo na velike opsege: ukoliko neki brojač mora da se nalazi između 1 i 9 napisaćemo nešto poput *type cifra = 1..9*

Daleko je zanimljivija linija koja opisuje tip *jezici*: iza znaka jednakosti smo u običnim zagradama, ispisali imena nekoliko programskih jezika razdvojenih zarezima. Promenljivoj *brojač* je nekoliko linija docnije, dodeljen tip *jezici* što znači da ćemo u programu moći da napišemo nešto poput *brojač :=fortran* ili *brojač :=paskal*. Vrlo je bitno da primetimo da *brojač* **nije** alfanumerička promenljiva: da smo u bejziku napisali *BROJAČ$="FORTRAN"*, radilo bi se o prostom dodeljivanju teksta. U paskalu, sa druge strane, govorimo o tipovima koji su sasvim ravnopravni sa *integer* ili *Boolean*: nema nikakve smetnje da napišemo *for brojač :=fortran to modula2 do* pa će *brojač* u petlji redom uzimati vrednosti imena svih programskih jezika.

Naredbe *for brojač :=fortran to modula2 do* nas uči još nečemu **vrlo** je bitno kojim smo redom naveli jezike. Sledili smo približnu hronologiju pojavljivanja što znači da će u našoj petlji brojač dobiti najpre vrednost *fortran*, zatim *cobol* pa onda *algol*, *bejzik*, *paskal* i na kraju *modula2*. Da smo, sa druge strane, napisali *for brojač :=modula2 to fortran do*, sadržaj petlje ne bi bio izvršen ni jedan jedini put pošto se podrazumeva da je "broj" *modula2* veći od "broja" *fortran*. Slika 6.3 opisuje još nekoliko primera kao što su *boje* i *kompjuteri*: nema nikakvih pro-


```
program liste (input, output);
type   gradj_ptr   =   ^gradjanin;
       gradjanin   =   record
                         sledeci  gradj_ptr;
                         lkarta   integer;
                         pasos    integer
                       end;
var    pocetni   gradj_ptr;
       dalje     Boolean;
       izbor     integer;
procedure insert (var pocetni  gradj_ptr;
                  lk,pa  integer);
var    zadnji   gradj_ptr;
begin
  if  pocetni = nil then
  begin
    new (pocetni);
    pocetni^.sledeci:=nil;
    pocetni^.pasos :=pa;
    pocetni^.lkarta:=lk
  end else begin
    zadnji :=pocetni;
    while zadnji^.sledeci<>nil do
      zadnji:=zadnji^.sledeci;
    new(zadnji^.sledeci);
    zadnji :=zadnji^.sledeci;
    zadnji^.lkarta :=lk;
    zadnji^.pasos :=pa;
    zadnji^.sledeci :=nil
  end
end;

procedure delete (var pocetni  gradj_ptr;
                  lk  integer);
var    zadnji,prethodni    gradj_ptr;
       dalje               Boolean;
begin
  zadnji:=pocetni;
  prethodni :=pocetni;
  dalje :=prethodni<>nil;
  while dalje do
    if zadnji^.lkarta = lk then begin
      dalje :=false;
      if  zadnji=pocetni
      then
        pocetni:=zadnji^.sledeci
      else
        prethodni^.sledeci :=zadnji^.sledeci;
        dispose(zadnji);
    end else begin
      prethodni :=zadnji;
      zadnji :=zadnji^.sledeci;
      if zadnji=nil
        then dalje := false
    end;
  if  zadnji=nil
  then
      writeln ('Podatak ne postoji!')
end;
begin
  pocetni := nil;
  { ... }
end
```

Slika 6.20

zovu pointeri, odlučili smo se da reč *pointer* koristimo kada govorimo o tipu a njen prevod (*ukazatelj*) kada govorimo o **promenljivoj**. Ukazatelji i liste su usko povezani: ukazatelj pokazuje na listu dok je lista određena jedino preko ukazatelja.

Svaka promenljiva u paskalu, dinamička ili ne, ima svoj tip koji se obično deklariše u okviru *var* sekcije programa. Liste ne mogu ovako da se deklarišu jer nemaju ime i jer je njihova veličina neodređena. Zato se opis dinamičkih struktura daje u okviru *type* sekcije kao u primeru 6.20

Deklaracije sa slike 6.20 uvode jednu listu čiji će elementi biti podaci o građanima: broj lične karte i broj pasoša. Osim ova dva broja svaki slog mora da sadrži i ukazatelj na sledeći slog u ulančanoj listi. Prvi slog će, dakle, ukazivati na drugi, drugi na treći i tako dalje, poslednji će slog ukazivati na "ništa" pri čemu se to "ništa" na paskalu zove *nil*. Šta nam nedostaje? Naravno, nešto treba da ukazuje na prvi slog pa smo uveli promenljivu *početni* tipa pointer

Jeste li primetili da primer 6.20 prikazuje nešto veoma neobično za paskal: tip *gradj_ptr* se definiše kao *^gradjanin* pri čemu će tip *gradjanin* biti definisan tek u sledećem redu: do sada smo smeli da koristimo isključivo strukture koje smo već definisali! Ovo se odstupanje od dobrih pravila strukturiranog programiranja na žalost nije moglo izbeći: da smo najpre definisali tip *gradjanin*, ne bismo mogli da opišemo jedno od njegovih polja koje je obavezno pointer tipa *gradj_ptr*!

Izvršni deo programa 6.20 dodaje elemente listi. Procedure je jasno, potrebna inicijalizacija u glavnom programu smo napisali *početni :=nil* što znači da je lista prazna. Kada poželimo da joj dodamo neki element, koristićemo ugrađenu proceduru *new*: posle *new (početni)*, računar će rezervisati prostor za prvi element liste a zatim ukazatelju *početni* dodeliti adresu tog slobodnog prostora. Sada možemo da dodeljujemo vrednosti *početni^.lična_karta*, na primer, označava polje *lična_karta* u zapisu na koga pokazuje *početni*

Procedura *delete* omogućava brisanje nepotrebnih elemenata liste. Treba najpre pronaći odgovarajući element a zatim modifikovati ukazatelj prethodnog elementa tako da pokazuje na sledeći. Tekući element, međutim, nije "mrtav" premda ništa ne pokazuje na njega: treba se poslužiti procedurom *dispose(x)* koja oslobađa prostor na koji je *x* ukazivao. Kompletan program za rad sa listama dat je u već pomenutom umetku "Sa bejzika na paskal"

U čemu su prednosti a u čemu mane dinamičkih struktura podataka? Glavnu prednost smo već pomenuli: rezerviše se samo onoliko memorije koliko je neophodno a to može biti i više od 64 K. Glavna je mana činjenica da elementima liste moramo da pristupamo sekvencijalno: da bismo pronašli hiljaditi, moramo da pročitamo 999 prethodnih! Rad se ponekad može ubrzati tako što će svaki slog imati više ukazatelja: na primer ukazatelj na prethodni i ukazatelj na sledeći slog. U programiranju se, osim toga, vrlo često koriste strukture nazvane **stabla**: liste je izuzetno zgodne za memorisanje stabla što znači da će razumevanje dinamičkih struktura podataka pomoći pri realizaciji složenijih algoritama

Naredba *exit* omogućava forsiran izlazak iz procedure i povratak u pozivajući program *exit* u glavnom programu ponaša se kao i *halt*

Ugrađene procedure

Kao što postoje ugrađene funkcije tako postoje i ugrađene procedure za koda ćemo upoznati *read readln write* i *writeln* dok ćemo se ostalim procedurama baviti kada nam zatrebaju

Na treba biti veliki poznavalac kompjutera da bi se zaključilo da *read* predstavlja učitavanje sa tastatura a *write* ispisivanje na ekran. Ono *ln* koje se ponekad "prikrpi" na *read* ili *write* je skraćenica od *line* razlika između procedura *write* i *writeln* je u tome što posle izvršavanja prve kurzor ostaje u letom redu (kao da smo na bejziku napisali PRINT "ABCD",) dok druga izaziva prelazak kurzora u sledeći red po završetku ispisivanja. Situacija kod *read* je nešto složenija što znači da je u praksi najbolje koristiti *readln*

Argumenti procedure *read* su promenljiva čije sadržaje treba učitati sa tastature Procedura *write* je nešto kompilkovanija njen argument pre svega, može da bude string pod apostrofima što se direktno prenosi na ekran. Ispisivanjem vrednosti promenljive se na neki način može upravljati a 4 na primer znači da će vrednost celobrojne promenljive u zauzimati četiri pozicije na ekranu ako je broj manji od 1000 biće dopunjen vodećim blankovima. Vrednost racionalne promenljive z može se ispisati se *z:10:3* što znači da će vrednost zauzimati deset mesta (u ih deset mesta ulaze decimalna tačka i predznak broja) pri čemu će se ispisati tačno tri decimala.

Skupovi

```
program skupovi,

type jezici      = (fortran, cobol, algol,
                    basic, pascal, module2);
     zna_jez     = set of jezici,

var  programer1 programer2,
     programer3, programer4,
     sveznalica, pecetnik      : zna_jez;

begin
   programer1 := [fortran, cobol, pascal],
   programer2 := [basic],
   sveznalica := [fortran .module2],
   pecetnik := [ ],
   programer3 := [algol pascal],
   { ... }
end.
```

Slika 6.19

Na engleskom se skup kaže *set* pa se i na paskalu tako definiše pogledajmo primer 6.19 Definisali smo kao i ranije tip *jezici* koji nabraja nekoliko popularnih programskih jezika, zatim tip *zna_jez* kao *set of jezici* i najzad nekoliko promenljivih tipa *zna_jez* Prve izvršne naredbe kažu da *programer1* zna fortran, kobol i paskal, da *programer2* zna samo bejzik da *sveznalica* poznaje sva jezika *e početnik* ni jedan jedini Sintaksa je maksimalno pojednostavljena elementi skupa se navode u srednjim zagradama (u matematici su uobičajena velika) i razdvajaju zarezima. Redosled elemenata u okviru skupa je lasno irelevantan.

Poslednja izvršna naredba programa 6.19 pokazuje da na moramo uvek da nabrajamo elemente skupa – možemo da napišemo *algol pascal* pa će se podrazumevati da *programer2* zna algol bejzik i paskal Proverite svoje znanja tako što ćete napisati da *programer4* zna sve jezike osim bejzika.

Paskal omogućava da operišete sa skupovima kao sa brojevima plus, na primer, označava uniju. Koliko je *programer1 + programer2* ili, u prevodu, koje sva jezika poznaju prvi i drugi programer? Odgovor je, jasno fortran..bejzik, kobol i paskal

Zvezdica označava presek skupova. *programer4 * programer2* bi izvršilo jezike koje znaju i prvi i drugi programer u našem je primeru to prazan skup. Unija i presek su komutativne operacije što znači da je sasvim svejedno da li će biti napisali *e+b* ili *b+e*

Minus označava razliku skupova. *programer3 - programer1* izdvaja sva jezika koje traći programer zna a koje istovremeno prvi na zna u našem su primeru to algol i bejzik Razlika nije komutativna sami pokažite da bi *programer1 programer3* rezultiralo skupom čiji su elementi fortran i kobol

Prisazimo na relacije Skupove možemo da poredimo kao i obične brojeve a tim što je smisao nekih operatora unikalno između. No, podimo redom.

x=y ima vrednost *true* ako su *x* i *y* jedneki skupovi na zaboravimo da redosled elemenata skupa nije bitan što znači da su skupovi [bejzik, paskal] i [paskal bejzik] jednaki.

x<>y je *true* ako skupovi *x* i *y* nisu jednaki.

x<=y je *true* ako je *x* podskup od *y* svi elementi skupa *x* su istovremeno i elementi skupa *y*

x>=y je *true* ako je *y* podskup od *x* svi elementi skupa *y* su istovremeno i elementi skupa *x*.

Ostalo je još da pomenemo relaciju *in* koja ima vrednost *true* ako je neki element u nekom skupu *fortran in programer1* ima vrednost true (jer prvi programer zna fortran) ali je zato *fortran in programer3* jednako *false*. Ne smemo da zaboravimo ni ograničenje Turbo Pascala skup može da ima najviše 64 elementa.

Rad sa fajlovima

Zbog ograničenja mikroprocesora 8086, Turbo Pascal može da razdvaja svega 64 kilobajta memorije za neku matricu ili niz ukoliko su potrebne veće strukture ili ukoliko se baš štednji memorije treba koristiti dinamičke strukture podataka liste. Elementi liste zauzimaju minimum prostora prostor za novi element se rezerviše tek kada je taj element zaista potreban da bi se čim element više na bude bio potreban prostor oslobodio za druge stvari.

Uvedimo najpre novi tip podataka pointere ili ukazatelje Da bismo ukratili zabunu koju često izaziva činjenica da se promenljive tipa pointer takođe

```
const  maxm     = 50,

type   indeks   = 0..maxm,
       matrica  = array [indeks] of real;
       jezici   = (fortran, cobol, algol,
                   bejzik, pascal, module2),

var    podatak  array [0 .190] of integer;
       pomeran  array [-199. 199] of real,
       se       array [indeks] of real;
       bb       matrica,
       cc       array [Boolsen] of integer;
       deskriptiv: array [jezici] of integer;
       brojac   jezici,
```

Slika 6.4

```
program deskrip (input,output),

type   jezici  =  (fortran, cobol, algol,
                   bejzik, pascal, module2),

var    brojac;     jezici,
       najveci     integer;

const  deskriptiv  array [jezici] of integer =
                   (90,116,199,75,109,105),

begin
   najveci :=0;
   for brojac = fortran to module2 do
      if deskriptiv[brojac] > najveci
         then najveci :=deskriptiv[brojac];
   writeln (najveci)
end
```

Slika 6.5

bilema da sami definišete baš ona tipova promenljivih koja su potrebne da biste rešili problem koji vam je postavljen umesto da kao u bejziku ili fortranu trošite energiju na simuliranje potrebnih tipova podataka definišite tipove koji su vam potrebni i trošite snagu na rešavanje problema! Ova sasvim ispravna koncepcija paskale nije, na žalost dotegla do kraja jer su ulazno/izlazne operacije se koriničiće definisanim tipovima praktično nemoguće!

Programiranje na fortranoidnim jezicima se jasno, na može zamisliti bez nizova i matrica koje ćemo upoznati na primeru 6 4 Prva deklaracija je najelićnija bejziku uvek smo niz celih brojeva *podatak* čiji indeksi uzimaju vrednosti od 0 do 100 baš kao da smo napisali DIM PODATAK%(100) Jedna od razlika odmah pada u oči u deklaraciji *podatak array [0 100] of integer* se pominju brojevi 0 i 100 dok se u bejziku navodi samo broj 100 pošto se podrazumeva da indeksi počinju od nule. Navođenje dodatnog podatka znači veću fleksibilnost. ako se nekade indeksi po svojoj prirodi budu nalazili između 100 i +100 napisaćemo *array [100 100]*

Treća deklaracija se slike 6 4 predstavlja varijaciju na temu umesto da dimenzionišući niz, u srednjim zagradama navedemo opseg indeksa definisali smo taj opseg kao tip *indeks* a zatim napisali *se array [indeks] of real* Da li je na ovaj način nešto dobijeno? Ukoliko u raznim potprogramima budemo koristili matrice istog tipa ušićedćemo malo pisanje i razmišljanja možda je još važnije što smo program učinili čitljivijim. Dalje unapređenje čitljivosti nudi deklaracija matrice *bb* definisali smo tip *matrica* kao *array [indeks] of real* a zatim napisali *bb matrica* što je sa stanovišta prevedenog programa ekvivalentno sa *bb array [0 50] of real*

Deklaracija *cc array [Boolean] of integer* je dobra prilika da proverimo naučeno. Znamo da logičke (ili Bulova) promenljive mogu da imaju vrednosti *true* i *false* što znači da paskal sadrži implicitnu deklaraciju *type Boolean = (false, true)*- Matrica cc će prema tome imati dva elemenia *cc [false]* i cc *[true]* pa ćemo im se tako obraćati u izvršnom delu programa Slično tome teorijski bismo mogli da napišemo *x: array [integer] of integer* i tako dimenzionišemo niz celih brojeva *x* čiji se indeksi nalaze između *-maxint* i *+maxint* Turbo Pascal međutim, neće moći da privati ovako veliku strukturu ali vas niko na sprečava da isprobate deklaraciju *x: array [0 maxint] of byte* Korišćenja nizova čiji su indeksi racionalni brojevi nije dopušteno što znači da je *hh array [real] of integer* nepohvatljiva deklaracija.

Sasvim je razumljivo da pri dimenzionisanju matrica niamo ograničeni osnovnim tipovima pogledajmo na primer matricu *deskriptiv* opisanu sa *deskriptiv: array [jezici] of integer* Sećamo se da je tip *jezici* opisan kao *jezici = (fortran cobol, algol bejzik paskal module2)* što znači de ćemo u izvršnom programu moći da napišemo neto poput *deskriptiv[paskal] =* 100 odnosno *deskriptiv[bejzik] –*90 ili da sastavimo procedure sa slike 6.5 koja prenesi najdeskriptivniji pregramski jezik.

Primer 6.5 je dobra prilika da upoznamo još jedno unapređenje Turbo Pascala u odnosu na ISO standarde – sekciju *const*. Standardni paskal obezbeduje samo skalarne konstante dok su u Turbo Pascalu konstante sasvim približene promenljivima što znači da mogu da budu i nizovi odnosno matrice - podatak o deskriptivnosti pojedinih jezika je dodeljen odgovarajućim konstantama u okviru *const* sekcija na način koji veoma podseća na DATA liste u fortranu

Ostalo je još da pomenemo višedimenzione matrice koje su, kao što se iz primera 6.5 vidi na paskalu rešene veoma fleksibilno Prvi red zvuči sasvim obično *x: array [0 maxx, 0 maxy] of real* je leto što I DIM X(MAXX MAXY) na bejziku. Drugi red je isto to samo malo drugačije *y: array [0 maxx] of array [0 maxy] of real* beš zvuči komplikovanol Matricu obično zamišljamo kao pravougaonu strukturu od *maxx*maxy* ćelija svaka ćelija sadrži po jedan na primer racionalan broj Matricu međutim možemo da zamislimo i kao niz od *maxx+1* kolona (zašto maxx+1? Zato što počinjemo od nule!) pri čemu se svaka kolona sastoji od *maxy+1* racionalnih brojeva Tu je beš i smisao definicije *x* je niz čiji se indeksi kreću od 0 do *maxx* Šta su elementi toga niza? To niau brojevi nego novi nizovi indeksi svakog od ovih nizova idu od 0 do *maxy* a elementi

```
const   maxx  = 50;
        maxy  = 50;
type    godine = 1980..1990;
        Dekart = -10..10;
        jezici = (fortran, cobol, algol,
                  bejzik, pascal, modula2);

var
   x: array [0..maxx, 0..maxy] of real,
   y: array [0..maxx] of array [0..maxy] of real,
   z: array [1..100, -12..12] of integer;
   verzija: array [jezici, godina] of godine;
   koordinate: array [Dekart, Dekart] of real,
```

Slika 6.6

niza su racionalni brojevi. Na sličan bismo način mogli da uvedemo i trodimenzione, četvorodimenzione i ostale nizove ali bi nas prečesto kucanje reči *array* svakako iznerviralo - zato je i omogućeno normalno navođenje indeksa u srednjoj zagradi što podseća na bejzik i fortran.

Nema posebne potrebe da svi indeksi višedimenzione matrice budu u istom opsegu, pogledajmo na primer matricu *z* sa slika 6.6 - prvi indeks ide od 1 do 100 a drugi od -12 do +12. Mnogo je međutim, zanimljivija matrica *verzija* koja opisuje verzije programskih jezika aktualne u raznim godinama u izvršnom delu programa može da se napiše naredba koja će glasiti *verzija [pascal, 1988] := 4* i čiji će smisao biti jasan čak i nekome ko nema pojma o programiranju.

Ostalo je još da pomenemo *pakovane matrice* koje su predviđena standardima paskala ali koje Turbo Pascal ne poznaje deklaracija *packed array* je sintaksno dopuštena ali se prevodi isto kao i *array*. Standardne procedure *pack* i *unpack* nisu implementirane.

Aritmetički i logički izrazi

Najčešći elementi izvršnog dela programa su aritmetički i logički izrazi. Počnimo od najjednostavnijeg izraza *a := b* kojim se vrednost *b* prenosi u *a*. Simbol ':=' je uveden da bi se dodeljivanje razlikovalo od poređenje. *a:=3* je naredba koja dodeljuje vrednost 3 promenljivoj *a* dok je *a=3* deo logičkog izraza kojim se ispituje da li je sadržaj promenljive *a* baš broj 3. Prenošenje vrednosti se ne ograničava samo na skalarne promenljive ali se uvek zahteva da izraz sa desne strane znaka jednakosti bude istog tipa kao i promenljiva se leve strane eko želiko da dodelimo racionalnu vrednost celobrojnoj promenljivoj moraćemo da koristimo funkciju *trunc*.

Za razliku od fortrana i bejzika paskal razlikuje operacije sa celim, racionalnim i Bulovim brojevima: sa celim brojevima "rade" operatori +, - *mod* i *div*, za racionalne su "zaduženi" +, - i / a za Bulove *and*, *or* i *not*. Pošto svi dobro znamo da plus označava sabiranje minus oduzimanje, zvezdica množenje kosa crta deljenje *and*, *or* i *not* konjukciju, disjunkciju i negaciju a *div* i *mod* celobrojno deljenje odnosno ostatak pri deljenju, ostaje nam samo da se pozabavimo specifičnostima Turbo Pascala.

Mora se reći da je Borlandov kompajler odstupio od "čiste" koncepcije paskala utoliko što je pomešao celobrojne i Bulove vrednosti operatori *and*, *or*, *not* i *xor* (ekskluzivna disjunkcija, još jedna ekstenzija Turbo Pascala) mogu da se koriste kao logički u kom slučaju operišu sa vrednostima *true* i *false* ili kao celobrojni u kom slučaju vrše logičke operacije bit po bit - ako je vrednost promenljive *a* 210 a *b* 111 (obe su promenljive primera radi deklarisane kao *byte*) *c := a and b* će promenljivoj *c* dodeliti vrednost 43 (znate li zašto?) Uvedeni su i dodatni celobrojni operatori *shl* i *shr* koji obezbeđuju aritmetičko šiftovanje na levo ili na desno za određeni broj mesta. na primer *a:=b shr 4*. Pomenućemo još i unarni operator @ kojim definišemo pointer na neku promenljivu ili proceduru (*npr vradpoint:=@vradnost*) što je naročito značajno kada se paskal programi povezuju sa asemblerskim potprogramima.

Relacioni operatori =, <>, <, >, <=, i >= deluju na način uobičajen u većini programskih jezika. Prioriteti su, međutim, pomalo čudno rešeni što najbolje ilustruje naredba *if a=b and c=d then* pošto su logički operatori *and* i *or* po prioritetu izjednačeni sa množenjem i sabiranjem računar će pokušati da izračuna *b and c* pa da zatim ispita da li je rezultat jednak *a*. Turbo Pascal nema ništa protiv toga ali će se pobuniti što se leko dobijeni razultat poređenja ponovo poredi sa celim brojem što je srećna okolnost - da je naredba proglašena za sintaksno ispravnu program bi radio sasvim pogrešno iz razloga koja bi malo koji početnik razumeo! Bilo kako bilo logički izrazi u paskal programima su uvek prepuni zagrada što znači da ćemo pomenuti naredbu pisati leo *if (a=b) and (c=d) then*. Ukoliko ste u dilemi oko prioriteta operacija, uvek je bolja da stavite zagrade - od njih šteta na može da bude!

Što se racionalnih brojeva tiče, na raspolaganju nam je sabiranje oduzimanje, množenje i deljenje ali, paradoksalno, ne i stepenovanje (treba znati da je *a**b = exp(b*ln(a))*). Tu su, naravno 1 elementarne funkcije koje smo ne slici 6.7 propratili kratkim opisima.

abs	Apsolutna vrednost
arctan	Arkus tangens
cos	Kosinus
exp	E**X
frac	Razlomljeni deo
int	Ceo deo (razultat je racionalan broj)
ln	Prirodni logaritam
pi	Vrednost broja π
random	Slučajan broj između 0 i 1. Ako se navede celobrojni argument *n* ceo slučajan broj iz intervala (0 n)
round	Zaokruživanje (razultat je *longint*)
sin	Sinus
sqr	Kvadriranje
sqrt	Kvadratni koren
trunc	Odsecanje decimala (razultat je *longint*)

Slika 6.7


```
program hanoj (input,output);

const   maxn  = 64;

type    diskovi = 1..3,
        raspon = 0..maxn;

var     n       integer;

procedure prenos (n:raspon; x,y,z: diskovi);
begin
   if n<>0 then
   begin
      prenos(n-1,x,z,y);
      writeln('Sa',x:3,' na',y:3);
      prenos(n-1,z,y,x)
   end
end;

begin  { glavni program }
   write ('n=');
   readln (n);
   if n>maxn then
      writeln ('Ceka ga beo kraje svetai')
   else prenos (n,1,2,3),
end
```

Slika 6.16

```
program demo,
{
Program koji upada u mrtvu petlju!!
}
var a,b  integer;

procedure q(var x: integer),
forward,

procedure p(var y: integer),
begin
   q(y)
end,

procedure q;
begin
   p(x)
end,

begin
   p(a),
   q(b)
end
```

Slika 6.17

Specijaliteti Turbo Pascala

Već smo videli da Turbo Pascalu nedostaje mogućnost prenošenje procedura kao parametara ali to ne znači da je on na potju potprograma inferioran standardnom paskalu. Borland je pripremio i nekoliko veoma interesantnih poboljšanja od kojih ćemo ovde pomenuti klauzule *interrupt*, *external* i *inline* i naredbu *exit*.

Najlakše je razumeti klauzulu *external* ako program pisan ne Turbo Pascalu treba da pozove neki potprogram pisan na asembleru taj potprogram treba na neki način uključiti u funsinu EXE datoteku obezbeđujući pri tome mehanizme za poziv i prenos parametara. Pošto u paskalu (skoro) sve što se koristi mora biti deklarisano definiciju *eksterne* procedura ćemo na primer zameniti sa.

```
procedure popuni  (var gde, podatak: longint,
                   koliko: longint),
external,
{$L POPUNI OBJ}
```

Pretpostavka je da je POPUNI OBJ datoteka sa asemblirnim programom koji preuzima vrednosti argumenata i popunjava blok memorije podacima. Ovakav prenos praktikujemo samo eko je korišćeni segment mašinskog programa nešto duži par asemblerskih instrukcija možemo da umetnemo as *inline* kao u sledećem primeru (tako ćete pogodili da je $FA kod instrukcije CLI koja onemogućava prekide)

```
procedure DisableInterrupt, inline ($FA),
```

Pomenimo i klauzulu *interrupt* koja omogućava definisanje interapstkih procedura. Interapstka procedura **uvek** mora da počinje sa

```
procedure ime (flags,cs,ip,ax,bx,cx,
               dx si,di,ds,es,bp  word),
interrupt;
```

Jeste li primetili nešto neobično u prethodnim primerima? Trebalo bi da jeste - skrećemo vam pažnju na konstrukciju *procedure popuni (var gde, podatak: longint; koliko: longint)*, u kojoj formalni parametar *gde* uopšte nije deklarisan. Nešto slično je na standardnom paskalu nezamislivo i sa te strane ovu ekstenziju Turbo Pascala nema mnogo smisla koristiti ponekad se ipak može ukazati potreba da napišemo proceduru koja će operisati se raznim tipovima podataka. Program 6.18 ilustruje poređenje dve strukture bez obzira na njihov tip uz pomoć procedure *equal*

```
program proba (input, output),
type vector = array [1..10] of integer;
     point  =  record
                  x, y:   integer;
               end,
var   vec1, vec2: vector;
      n:          integer;
      p:          point,
      rez1, rez2,
      rez3:       Boolean,
function equal (var source, dest;
                size: word): Boolean;
type bytes =  array [0..maxint] of byte,
var   n:      integer,
begin
   n := 0;
   while (n< size) and
   (bytes(dest) [n] <> bytes(source)[n]) do
      inc(n),
   equal := n = size
end,
begin
   rez1 := equal(vec1, vec2, sizeof(vector));
   rez2 := equal(vec1, vec2, sizeof(integer)*n);
   rez2 := equal(vec1[1], p, 4)
end
```

Slika 6.18

```
program testiranje (input,output),

{ pogodite rezultate! }

var y,z,i,j integer;

procedure test1 (var e integer),
var i,j integer;
begin
   i:=4, j =7,
   z:=3; e =5,
end,

procedure test (a integer);
var i,b,z: integer;
begin
   test1 (i),
   writeln ('i= ', i:5);
   z:=44,
   i =9; j:=999,
   test1 (b),
   writeln ('b= ', b 5),
   writeln ('z= ', z:5),
end,

begin { glavni program }
   z:=66, i =96, j =96,
   y:=363;
   test(y),
   writeln ('z= ',z:5),
   writeln ('y= ',y.5),
   writeln ('i= ',i 5),
   writeln ('j= ',j 5)
end
```

Slika 6 14

Turbo Pascal 4 0 čak omogućava da neke procedure budu upisane u odvojene datoteke na način koji ćemo upoznati u sledećem poglavlju.

Slika 6.14 treba da vam omogući da proverite stečeno stanje pokušajte da predvidite rezultate izvršavanja programa a zatim ga startujte i proverite svoje zaključke Ukoliko se pojave razlike pokušajte da pronađete greške u sopstvenom razmišljanju tako se najbolje uči!

```
program faktorijel (input,output),
const   maxn   =12,
var     n'        integer;

function fakt (n. integer) longint,
var i,y' longint;
begin
   y'=1,
   for i :=2 to n do
   y' :=y*i,
   fakt:=y
end;
begin { glavni program }
   write ('n='),
   readln (n),
   if n>maxn then writeln ('Preveliko n!')
   else writeln (n.3,'! =',fakt(n)),
end
```

Slika 6 15

Funkcije su prilično slične procedurama može se reći da je funkcija specijalna procedura koja ima nekoliko ulaznih i tačno jednu izlaznu veličinu. Navikete ćemo razume se objasniti na primeru

Program 6.15 obavlja nešto veoma uobičajeno za početne kurseve raznih programskih jezika računa faktorijel Koncentrišimo se na prvi novitet, *function fakt (n integer) longint Function* je službena reč, *fakt* je ime funkcije *n* je argument (i to celobrojan zbog onog *integer*) dok je reč *longint* iza zatvorene zagrade neophodna jer svaka funkcija vraća jednu vrednost pa moramo da kažemo kog je te vrednost tipa u našem slučaju faktorijel je ceo broj pa smo napisali da je funkcija *fakt* celobrojna tj da vraća *longint*

Iza imena funkcije pronalazimo deklaracije lokalnih promenljivih *i* i *y* (funkcije mogu da imaju lokalne promenljive, lokalne konstante pa čak i lokalne funkcije i procedure) i na kraju *begin - end* strukturu koja računa faktorijel Primetimo, ipak, da se funkcija završava sa *fakt =y* na taj način stavljamo kompajleru na znanje da je vrednost koja treba da se vrati u glavni program *y* a ne *i* ili možda *y+1* vrednost funkcija se, dakle, na kraju obavezno dodeljuje njenom imenu Iako znamo da nas nećete poslušati, savetujemo vam da nikada ne pokušavate da "uštedite" promenljivu *y* koristeći ime funkcije kao "običnu" promenljivu sličan postupak predstavlja pravo pravcato traganje za nevoljama!

Zanimljivo je da standardni paskal omogućava da imena procedura i funkcija budu argumenti drugih procedura i funkcija ali je nešto slično na Turbo Pascalu nemoguće - ovo je jedna od retkih situacija u kojoj je delo Borlandovih programera inferiorno u odnosu na većinu ostalih paskal kompajlera.

Rekurzije su dozvoljene kako na paskalu tako i na Turbo Pascalu – procedura ili funkcija može pozvati samu sebe pri čemu se programer mora potruditi da izbegne mrtvu petlju Primer 6 16 rešavajući vekovni problem zvani "Kule Hanoja", ilustruje upotrebu rekurzije. "Kule Hanoja" su, za one koji to još ne znaju zasnovane na legendi da u gradu Benaresu postoji podzemni hram u kome je induski bog Brama pri stvaranju sveta postavio tri dijamantska štapića i na jedan od njih stavio 64 zlatna kotura različitih veličina na zemlju je stavio najveći kotur, na njega manji i tako sve do najmanjeg kotura koji se nalazi na vrhu tajanstvene kupe Stotinu sveštenika danju i noću prenosi koturove sa prvog na drugi štapić služeći se trećim kao pomoćnim i poštujući ograničenje koje je zadao Brama prsteni smeju da se premeštaju samo jedan po jedan i nikada veći prsten ne sme da se stavi preko manjeg. Kada sveštenici budu obavili ovaj na prvi pogled jednostavan posao nastupiće kraj sveta zbog čega se ne treba previše brinuti jednostavna računica pokazuje da nam je do kraja sveta ostalo bar još 500 milijardi godina.

Neki rekurzivni algoritmi zahtevaju da procedura A poziva proceduru B a procedura B proceduru A što za sada ne umemo da uradimo paskal gotovo bez izuzetka zahteva da ono što koristimo obavezno bude već definisano. Ovaj se problem rešava kao na slici 6.17 korišćenjem službene reči *forward* napišemo ime funkcije i njene argumente a zatim, umesto izvršnog dela, samo *forward* Ostaje nam, jasno obaveza da negde pre glavnog programa dopišemo radni deo procedure.

Vidimo da je bejzik, kada se radi o funkcijama, bogatiji od paskala brojne bejzik funkcije nemaju ekvivalente što znači da ih treba zameniti odgovarajućim izrazima pri čemu moramo da obratimo pažnju na činjenicu da matematički savršeno korektan izraz nije uvek i numerički tačan. Tabela 6.7 da budemo pošteni, pokazuje da paskal ima jednu funkciju koju bejzik ne poznaje ali bi, neka nam Virt oprosti bilo mnogo bolje da ni nju nema na bejziku je SQR kvadratni koren dok je na paskalu *sqrt* kvadratni koren dok *sqr* diže argument na kvadrat. Ukoliko, dakle, nekada budete pisali paskal program koji projektuje most pa se most docnije sruši (10 mrtvih 200 ranjenih) proverite da niste pisali *sqr* umesto *sqrt*!

S vremena na vreme se, jasno, mora raditi sa promenljivima koje nisu ni celobrojne, ni racionalne ni logičke. Tada ćemo se poslužiti funkcijama *ord*, *chr*, *pred* i *succ* prva daje poziciju nekog elementa u skupu, druga element na osnovu pozicije a preostale dve prethodni odnosno sledeći element skupa. Sve je mnogo lakše razumeti na primeru. pretpostavili da smo kao i par puta ranije definisali tip *boje* sa *type boje = (crna, bela, crvena zelena, žuta)* "brojevi" *crna, bela, crvena, zelena* i *žuta* su sortirani u rastući poredak pa će *ord (crna)* dati vrednost 0 (prvi odnosno nulti "broj") dok će *ord (žuta)* biti četiri. Sasvim slično tome, biće *succ (zelena)* – *žuta* i *pred (zelena)* = *crvena* Kada se radi o alfanumericima (*char*) *ord* daje ASCII kod karaktera a *chr* pretvara ASCII kod u karakter - baš kao u bejziku! Nekoliko ilustracija ovih funkcija nudi slika 6.8.

```
program karakteri (input,output),

{ ispisivanje ASCII seta }

var  znak   char;
     i.     32..126,

begin
   writeln ('ASCII set '),
   for i := 32 to 126 do
   begin
      znak =chr(i),
      write (znak),
   end;
   writeln,
   repeat
      write ('Unesi znak '),
      readln (znak),
      i.= ord (znak),
      writeln ( ASCII kod znaka ',znak, je ',i 6)
   until znak='@'
end
```

Slika 6 8

U aritmetičkim izrazima javljaće se brojne konstante koje se zadaju prema standardima paskala. Racionalni brojevi se razlikuju od celih po tački i/ili eksponentu pa ćemo na primer, broj $1.2*10^{12}$ pisati kao 1.2E12 ili, ekvivalentno, 1 2e12. Turbo Pascal dopušta i zadavanje celobrojnih konstanti u heksadekadnom brojnom sistemu za šta se koristi prefiks $ na primer $A1 ili $FFFFFFFF

Kontrolne strukture

U paskalu postoji jedna na prvi pogled veoma čudna konvencija bilo koja procedura funkcija, petlja ili neka druga struktura uključujući i sam glavni program se sastoji od **jedne jedine naredbe** Jedna naredba je, jasno sasvim nedovoljna da bi se obavilo bilo šta ozbiljno pa se umesto neke izvršne rečenice koristi konstrukcija *begin end* Kada paskal kompajler analizira neku strukturu i očekuje izvršnu naredbu, pronađeno *begin* će izazvati izvršavanje svih naredbi do odgovarajuće naredbe *end*

Osim sekvence, strukturirano programiranje zahteva ispitivanje i petlju Ispitivanje se realizuje primenom konstrukcije *if then else* ukoliko logički izraz iza *if* ima vrednost *true* biće izvršena **jedna** naredba koja se nalazi iza *then* dok će u slučaju da izraz ima vrednost *false* biti izvršena **jedna** naredba iza *else* (else sekcija, jasno može i da se izostavi) Umesto jedne naredbe možete da napišete reč *begin* i, iza nje, čitavu sekvencu naredbi (u kojoj smeju da se nalaze i nova ispitivanja) koja se završava sa *end*

Paskal podržava tri vrste petlji *while do*, *repeat until* i *for do* Prva petlja se smatra najuniverzalnijom iza *while* se piše neki logički uslov a iza *do* jedna naredba (ili *begin end*) koja se izvršava **sve dok je uslov ispunjen** ukoliko uslov u startu nije ispunjen petlja se neće izvršiti ni jednom *Repeat until* petlja je nešto drugačija izvršava se **sve dok uslov iza until ne bude ispunjen** Primetimo da se sadržaj *repeat until* petlje izvršava bar jednom dok se naredbe iza *while do* ne mora uopšte izvršiti pa *repeat - until* zovemo "petlja sa izlazom na dnu" a *while do* "petlje sa izlazom na početku" Obzirom da reči *repeat* i *until* na neki način uokviruju sekvencu, ne moramo da koristimo *begin end* strukturu premda ona, jasno neće biti na smetnji.

Ostala je još brojačka petlja *for brojač =početna to krajnja do* Brojačka promenljiva može da bude bilo kog skalarnog tipa osim *real* (pod *real* ovde podrazumevamo i različite racionalne promenljive koje podržava aritmetički koprocesor na primer *extended*) a priraštaj je uvek jedan pri čemu se može zahtevati "brojanje u nazad" sintaksa je tada *for brojač =početna downto krajnja do*

Primer 6.9 ilustruje strukture *case* koja predstavlja dobru zamenu za seriju sukcesivnih *if*-ova Uz pomoć ove strukture možemo da definišemo ponašanje programa u zavisnosti od neke kontrolne promenljive svaka od unetih vrednosti (predviđeno je veliko i malo slovo!) rezultira izvršavanjem jedne ili (uz *begin - end*) nekoliko naredbi Ukoliko promenljiva nema ni jednu od vrednosti predviđenih u okviru *case* strukture Turbo Pascal će izvršiti naredbu iza *else* što je unapređenje u odnosu na standarde paskala koji ne preciziraju ponašanje programa u sličnim situacijama. Neki drugi kompajleri poznaju klauzulu *otherwise* koja je funkcionalno ekvivalentna pomenutom *else*

I pored opšteg uverenja da je paskal nastao kada je instrukcija *goto* okrivljena za nepregledne i nepouzdane programe *goto* naredbe je element kako standardnog tako i Turbo paskala. Turbo Pascal je čak uneo i određena poboljšanja labela

```
program meni (input,output);
var   izbor  char;
      dalje  Boolean;
begin
   dalje := true;
   while dalje do
   begin
      writeln;
      writeln ('A  Unosenje podataka');
      writeln ('B  ispisivanje podataka');
      writeln ('C  Brisanje podataka');
      writeln ('D  Testiranje modela');
      writeln ('K  Kraj rada');
      writeln;
      write  ('    Vas izbor? ');
      readln (izbor);
      case izbor of
       'A','a' : writeln('Treba uneti podatke');
       'B','b' : writeln('Podaci se ispisuju');
       'C','c' : writeln('Brisanje dela podataka');
       'D','d' : writeln('Testiranje modela');
       'K','k' : dalje := false;
      else
              writeln ('Neispravan ulaz');
      end
   end
end
```

Slika 6.9

```
program test_proc(input,output);

const   maxn   = 100;

type    niz    = array [1 .. maxn] of integer;

var     x,y:           niz;
        poc, kraj,i,j  integer;

procedure ispis (e  niz; prvi,zadnji integer);
begin
   i := prvi;
   while i <= zadnji do
   begin
      writeln(e[i]) ;
      i := i + 1
   end;
   prvi := 0;
end;

begin  { glavni program }
   for j := 1 to maxn do
   begin
      x[j] := j;
      y[j] := -j
   end;
   write  ('Od kog se elementa ispisuje? ');
   readln (poc);
   write  ('Do kog se elementa ispisuje? ');
   readln (kraj);
   writeln ('Elementi niza X ');
   ispis (x,poc,kraj);
   writeln ('Elementi niza Y ');
   ispis (y poc,kraj);
   writeln ('poc=',poc:4)
end
```

Slika 6.10

se i dalje moraju definisati all se mogu sastojati i od slova a ne samo od cifara. *Goto* naredba je uglavnom ograničena na tekući blok - nisu dopušteni skokovi u susedne procedura dok je iskakanja u snažnu neke petlje strukturno dopuštano ali nepreporučljivo pošto se tada ne može predvideti ponašanje programa. Naredbu *goto* sve u svemu, treba izbegavati.

Naredbe (ispravnije bi bilo reći procedura) *halt* predstavlja ekvivalent bejzik instrukcije STOP nalazak na *halt* prekida izvršavanje programa i predaje kontrolu operativnom sistemu. Standardni paskal ne poznaje naredbu *halt* što znači da se izvršavanje prekida jedino kada se nalde na finalni *end*.

Procedure i funkcije

Već smo upoznali mnoge funkcija predvidene paskal standardom *trunc, abs sin sqrt, ord succ* upoznaćemo još poneku kada nam buda zatrebala. Funkcije se, dakle pozivaju pratim navodenjem imena i lista argumenata u zagradi e programeru je pružene mogućnost da definiše svoje funkcije i procedure koje su potpuno ravnopravne se firmirima pa se i pozivaju na isti način.

Želja da se funkcija poziva pratim navodenjem imena može da izazove jedan problem kako će kompajler razlikovati poziv funkcije od pristupanja elementu matrice? Nije bilo druge nego uvesti dva nova specijalna znaka *dok [=test(j)* predstavlja poziv procedura *test* čiji je argument *i* / *j =matrix[i* prenasi vrednost *i*-tog elementa niza *mat* u promenjivu *j*.

Slika 6.10 prikazuje sasvim jednostavnu proceduru koja ispisuje "komadić" niza A. Službena reč *procedure* označava početak potprograma. *Ispis* je njegovo ime dok su u zagradama navedeni formalni argumenti. Procedura *ispis* ima tri formalna argumenta: *e, prvi* i *zadnji*. *e* je niz a *prvi* i *zadnji* celobrojne promenljive. Zašto ove promenljive nazivamo *formalnim* argumentima? Spustimo pogled par redova niže i pročitajmo naredbu koja je procedura pozvane *ispis (x poc kraj)*. Paskal kompajler će naćpre pronaći proceduru a onde uporedit argumente (sl. 6.11) koji moraju da budu saglasni po vrsti i redoslednu: niza *x* odgovara niz *e* celobrojnoj promenljivoj *poc* celobrojne promenljive *prvi* e celobrojnoj promenljivoj *kraj* celobrojne promenljiva *zadnji*. Vrednosti stvarnih argumenata se dodeljuju formalnima a onda se izvršava procedura. Korist od argumenata vidljiva je iz glavnog programa - proceduru *ispis* smo ponovo pozvali all ovoga puta sa drugim argumentima (*y 5, 10*) i sve je bilo u najboljem redu ne standardnom bejziku ovakvo višestruko korišćenje istog potprograma zahtiva gomilu dodatnih LET naredbi.

```
e     prvi    zadnji   - fiktivni argumenti
x     poc     kraj     - stvarni argumenti

e     prvi    zadnji   - fiktivni argumenti
y     5       15       - stvarni argumenti
```

Slika 6.11

```
program probe(output);
var a,b,c,d,x,y: integer;

procedure niska (var a,b,c,d  integer,
x,y:    integer);
begin
   a:=5, b:= 1, c:=2; d:=3,
   x:=4, y:=5
end;

begin
   a :=7,  b :=5,  c:= 11; d :=13;
   x := 15, y:= 17,
   writeln ('Pre ulaska u proceduru:'),
   writeln ('a=',a:5,',b=',b:5,',c=',c:5),
   writeln ('d=',d:5,',x=',x:5,',y=',y:5),
   niska (a,b,c,d,x,y),
   writeln ('Po izlasku iz procedure.'),
   writeln ('a=',a:5,  ,b=  ,b:5,',c=',c:5),
   writeln ('d=',d:5,',x=  ,x:5,',y=',y:5),
end

Pre ulaska u proceduru
a=    7,b=    5,c=  11
d=   13 x=   15,y=  17
Po izlasku iz procedure:
a=    0,b=    1,c=   2
d=    3,x=   15,y=  17
```

Slika 6.12

Poito smo razumeli smisao, obratimo malo pažnje ne sintaksu. Formalni argumeti se navode u zagradi, razdvojeni tačkom i zarezom. Svaki argument mora da se deklariše na način koji smo već upoznali dvotačka i oznaka tipa. Samo se po sabi razuma da mimo ograničeni samo na skalarne tipove u programu 6.10 prenosili smo jedan niz a ne bi bilo problema ni da prenesemo neku promenljivu tipa *jezici* ili *boje*. Ukoliko prenosimo više promenljivih istog tipa njihova imena možemo da razdvojimo zarezima - baš kao kod var sekcije programa. Stvarni argumenti se, kao i u fortranu i modernijem bejziku razdvajaju zarezima.

Ukoliko se u toku izvršavanja procedure utiče na neki argument prenesen *po vrednosti* (npr. argument *prvi* se slike 6.10), ova će izmena po povratku u pozivajući program biti anulirana. Da bi procedura promenila vrednost nekog parametra, ovaj mora biti prenesen *po imenu* što se u paskalu označava navođenjem službene reči *var* kao na slici 6.12.

Procedura *niška* se slike 5.12 kao što joj i ime kaže ne radi ništa pametno zadatak joj je samo da menja vrednosti formalnih argumenata. Glavni program se druge strane, dodaljuje početne vrednosti i, po izvršanju procedure ispisuje rezultate- neke su vrednosti promenjene a neke nisu. Kako to? Odgovor pruža pogled ne deklaraciju procedura promenljive *a, b, c i d* su opisane kao *var a* promenljive *x i y* nisu potprogram je mogao da promeni vrednosti prvih all ne i drugih. Ukoliko dakle procedura ima neke lokalne veličine (tj promenljive koje vraćaju rezultate iz procedure u glavni program) odgovarajuće formalne argumente deklarišemo sa *var* ukoliko su promenljive isključivo ulazne (tj. služe za prenos podataka iz glavnog programa u proceduru) reč *var* izostavljamo. Korišćenje ulazno-izlaznih argumenata (tj promenljivih koje će donositi podatke i preuzimati rezultate) nije baš u duhu strukturiranog programiranja, ukoliko ipak ne težite za tim da budete "u duhu", i ove date promenljive morati da deklarišete se *var*.

```
program probe(output);
var a,b,c: integer;

procedure promena (var x: integer; y: integer);
begin
   x := 5; y := 5;
   c := 303
end;

begin
   a := 1,  b := 2; c := 3,
   writeln ('Pre ulaska u proceduru:');
   writeln ('a=',a:5,',  b=',b:5,',  c=',c:5);
   promena (a,b);
   writeln ('Po izlasku iz procedure:');
   writeln ('a=',a:5,',  b=',b:5,',  c=',c:5)
end.

Pre ulaska u proceduru
a=    1,  b=    2,  c=    3
Po izlasku iz procedure:
a=    5,  b=    2,  c=  303
```

Slika 6.13

Na slici 6.13 vidimo kratak paskal program i rezultate njegovog izvršavanja koje ćemo pokušati da prokomentarišemo. Definisali smo celobrojne promenljive *a, b* i *c* a onda im dodelili vrednosti 1, 2 i 3 respektivno. Sledi pozv procedura *promena* čiji su stvarni argumenti *a* i *b* a formalni *x* i *y*. U okviru procedura formalnim argumentima *x* i *y* dodeljujemo vrednost 5 all je po povratku u glavni program samo a promenjeno - *b* je, primećujemo, preneto *po vrednosti*. Do sada niemo primetili ništa novo. Pogledajmo, medutim, promenljivu *c* ona uopšte nije prenela u proceduru *promena* a opet je procedura *promena* uzela slobodu da promeni vrednost *c*; nova je vrednost štaviše, "preživela" i kraj izvršavanja procedura!

Procedura, dakle meše da analizira i menja vrednost promenljivih koje nisu prenete kao argumenti takve promenljive nazivamo *globalnim*. Rad sa globalnim promenljivima je u paskalu moguć all ne i preporučljiv - program je mnogo pregledniji i lakši za testiranje ukoliko su sve potrebne vrednosti prenesene preko liste argumenata.

Svaka procedura predstavlja paskal program u malom što znači da može (i treba) da ima lokalne konstante, tipove i promenljive koje važe samo u proceduri koja ih je definisala i eventualno u procedurama koja ta procedura dalje poziva. Lokalne promenljive mogu da imaju jedinstvena imena ili da nose imena nekih promenljivih glavnog programa ili drugih potprograma: neće doći ni do kakve zabune

Dejanove pitalice

Klikeraška pitalica

Prolećna tema, letnja pitalica i junski (tj. pomalo ispitni) odziv — sve u svemu, tačno 222 pisma i 220 tačnih odgovora dobijanih na nekoliko suštinski različitih načina. Prvu nagradu je, baš kao i prošlog meseca, zaslužio program za „komodor 128" — da li su se to „komodorci" pretplatili na najbrža rešanja?

Pitalicu ćemo, kao i obično, najpre ponoviti ali ćemo je tokom ponavljanja i rešavati — pravo je vreme da predstavimo matematičko (tj. nekompjuterizovano) rešenje do koga je nezavisno došlo mnogo naših čitalaca. Pera, Mika i Laza imaju po sto što zelenih što crvenih klikera, pri čemu svako od njih ima bar po jedan zeleni i bar po jedan crveni kliker. Broj Perinih, Mikinih i Lazinih crvenih klikera obeležićemo sa C1, C2 i C3, a odgovarajuće količine zelenih klikera sa Z1, Z2 i Z3. Prvi uslov kaže da je Ci+Zi=100 za i=1, 2, 3.

Pera je dao Miki onoliko crvenih klikera koliko je (Pera) imao zelenih — stanje je očito:

Pera C1−Z1 Z1
Mika C2+Z1 Z2

Zatim je Mika dao Lazi upola onoliko zelenih klikera koliko je Pera tada imao crvenih, što nas dovodi do stanja:

Mika C2+Z1 Z2−(C1−Z1)/2
Laza C3 Z3+(C1−Z1)/2

Najzad je Laza dao Peri onoliko zelenih klikera koliko je Mika tada imao crvenih, što nas je dovelo do stanja:

Pera C1−Z1 Z1+C2+Z1
Mika C2+Z1 Z2−(C1−Z1)/2
Laza C3 Z3+(C1−Z1)/2−(C2+Z1)

Posle ove komplikovane transakcije Pera i Mika su imali jednako crvenih klikera (C1−Z1=C2+Z1), a Laza je imao za jedan više crvenih nego zelenih klikera (C3=Z3+(C1−Z1)/2−(C2+Z1)+1). Da su svi Mikini klikeri bili zeleni, on bi imao jedan zeleni kliker više od Pere (C2+Z1+Z2−(C1−Z1)/2=Z1+Z1+C2+1). Uvođenjem smene Zi=100−Ci sistem se transformiše u tri jednačine sa tri nepoznate čijim rešavanjem dobijemo da su Pera, Mika i Laza imali 83, 49 i 34 crvena klikera, odnosno 17, 51 i 66 zelenih. Zar nije bilo jednostavno?

Zadatak se naravno može rešavati i kompjuterski — na slici 1 vidimo program zasnovan na principu grube sile, uz pomoć koga je naša „zvanična mašina" pronašla rešenje za 10 minuta i 1 sekund, prilično dugo ali se može sačekati! Slika 2 prikazuje mnogo bržu verziju zahvaljujući najboljoj ravnoteži između dužina programa i vremena izvršavanja donela prvu nagradu *Dianu Bećiroviću* iz Novog Beograda: program se na zvaničnoj mašini izvršio trenutno pa smo dopisali jedno FOR I=1 TO 1000 i tako došli do (i dalje zanemarljivih) 7 sekundi. Drugu i treću nagradu smo izvukli (primljeno je 222 tačna i dva pogrešna rešenja u kojima je dokazano da se zadatak ne može rešiti) — 40000 dinara je pripalo *Željku Barbiću* iz Zaboka, a 20000 dinara *Uzeiru Jusupoviću* iz Sarajeva. Pohvale za brze programe, vredna rešenja i zanimljive komentare zaslužuju *Milena Žolnaj, Bojana Miloradović, Milomir Aleksić, Davorin Jurišić, Krešimir Kos, Aleksandar Kragl i Petar Mitić.*

Rešenja zajedno sa programima koje ste koristili da ih dobijete šaljite na adresu „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd, tako da pristignu pre 25. avgusta 1988. Sva pisma sa korektnim rešenjima konkurišu za novčana nagrada od 60 000, 40 000 i 20 000 dinara, dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača pitalica. Identifikacioni broj dobijate tako što u prve tri kućice upišete *poslednje* tri cifre načijeg broja telefona, u sledeće dve godine vašeg rođenja i na kraju dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj.

```
REM
REM        KLIKERASKA PITALICA
REM
REM  Prema programu Zeljka Mosiska
REM
REM          Racunari 41
REM
CLS
PRINT TIMES
FOR m = 1 TO 99
    FOR 1 = 1 TO 99
        FOR p = 50 TO 99
            pc = 2 * p - 100
            mc = m  p + 100
            IF pc = mc THEN
                lz = 100 - 1 + pc / 2  m
                IF l = lz + 1 THEN
                    mz = 100 - m - pc / 2
                    pz = 100  p + mc
                    IF mc + mz = pz + 1 THEN
                        PRINT , " Crveni  Zeleni
                        PRINT "Pera " p 100 - p
                        PRINT "Mika " m 100  m
                        PRINT "Laza " 1 100  l
                        PRINT TIMES
                    END IF
                END IF
            END IF
        NEXT p
    NEXT 1
NEXT m
PRINT TIMES
END
```

```
REM
REM        KLIKERASKA PITALICA
REM
REM  Prema programu Diana Becir   ca
REM
REM          "Racu    41
REM
INPUT "Po koliko klikera imaju"
q = s / 2 + 1  r = s  q
y = (s  z) / 3  x = s  y
p = (s + (x  y  2) / 2) / 2  r = s  
IF q <> INT(q) OR p  INT(p) OR y    1  THEN
   PRINT "Resenje ne postoji"
ELSE
   PRINT , " Crveni   Zeleni"
   PRINT "Pera   x  y
   PRINT "Mika   z  q
   PRINT "Laza   r  )
END IF
END
```

Čikago, brada i orhideje

Jesu li vam dosadile lake pitalice? Nije da vas plašimo, ali 34 zadatak je našto teži. Dakle:

U malom avionu koji je putovao od Čikaga do Dalasa našla su se tri čoveka: Džejson, Kermit i Luis. Dvojica su članovi Mense, dvojica nose brade, dvojica gaje orhideje a dvojica su iz Čikaga. Nikoga na karakteriše više od tri pomenute osobine.

Ako je Džejson član Mense, on je iz Čikaga.

Ako Kermit nosi bradu, on gaji orhideje. Isto se može reći i za Luisa.

Ako je Džejson iz Čikaga, on gaji orhideje. Isto se može reći za Luisa.

Pitamo se ko je rođen u Dalasu. Da ne biste krenuli pogrešnim putem, zajednički ćemo proučiti rečenicu: Ako je Džejson iz Čikaga, on gaji orhideje *— odavde se ne sme zaključiti da Džejson koji (na primer) nije iz Čikaga ne* gaji orhideje!

Q zadatak

RQ9. Koja žena uvek tačno zna gde joj se muž nalazi? Ovo je bio lakozvani Q crnjak — udovica.

Q10. Koliko je sati kada se velika i mala kazaljka analognog časovnika poklapaju, a obe su na sredini između 1 i 2? Na prvi pogled bi se rekla da je 1 sat i sedam i po minuta ali...... (zadatak je predložio Vlada Paunović *iz Lazarevca)*

34: U Dalasu je rodjen ______________________

Ime i prezime ______________________

Adresa ______________________

Mesto ______________________

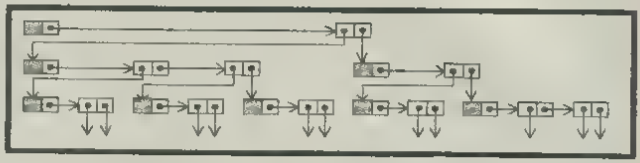
Kompjuter ______________ Vreme ______________

Identifikacioni broj: |__| |__| |__| (iz broja tel.) |__| |__| (god.rodj.) |__| |__| (po izboru)

Koren, stablo, list

U dosadašnjim nastavcima ste upoznali osnovna algoritma za rad sa listama. Videli ste kako se popunjavaju rupe nastale brisanjem elemenata liste i kako se u popunjavanju rupe mogu efikasno iskoristiti specijalne liste: stekovi, repovi i dvostruki repovi. Od ovog nastavka počinje upoznavanje sa efikasnijim strukturama podataka koje omogućavaju mnogo brži rad sa skupovima na računaru. Radi se, naravno, o stablima.

Stablo je, posle liste najubojitija programersko oružje iako postavlja neka restrikcija u vezi podataka sa kojima se može raditi i dosta komplikuje algoritme stablo je i dalje vrlo opšte struktura velike fleksibilnosti i primenljivosti Dovoljno je reći da svi kompajleri viših programskih jezika više od sedamdeset procenata svog vremena provode obrađujući razne stabla. Zajedno sa listama stabla čine okosnicu programskih jezika na kojima se razvija takozvana veštačka intaligencija.



Slika 1 — Razgranata lista

Šta je stablo

Praciznu matematičku definiciju stabla možete naći u RA 30/55 pa ćamo se ovde zadržati na rekurzivnoj definiciji koju od savremenih autora posebno forsira Niklaus Wirth Ona bi programerima trebalo da bude bliže od matematičke naravno ako su naučili da razmišljaju rekurzivno što i nije baš jednostavan posao U razumevanju definicije stabla može pomoći definicija liste budući da sa listom već umeia da operišete

Lista osnovnog tipa **T** (u dosadašnjim nastavcima je tej tip bio **Atom**) je

1 — prazna lista (tj nema nijednog elementa)
2 — spoj elementa tipa T i liste osnovnog tipa **T.**

Ovom je definicijom obuhvaćena sve prazna lista lista od jednog dva tri pa i beskonačno mnogo elemenate tipa T Jednostavno zar ne?

Stablo osnovnog tipa T (i dalja će to biti **Atom**) je

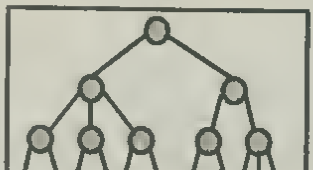
1 — prazno stablo
2 — element tipa T sa konačnim brojem podruženih, odvojanih stabala osnovnog tipa **T (podstabla)**

Stablo je za razliku od liste razgranata struktura Dok se kod lista sa jednog čvora uvek ide na samo jedan sledeći čvor kod stabla se sa jednog čvora prelazi na konačan broj čvorova Ukoliko je ovaj broj jednak jedinici kaže se da je stablo degenerisalo u listu (zbog toga neki autori listu neopravdano nazivaju degenerisanim stablom)

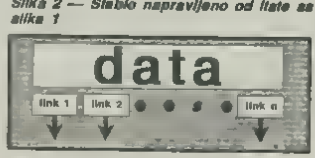
Na osnovu do sada rečenog moglo bi se zaključiti da je stablo toliko složenija struktura od liste da se pomoću lista nikako ne bi moglo načiniti stablo Kad bi to bilo tačno lista bi s pravom nazivali degenerisanim stablima Pogledajte međutim liste na sledećoj slici 1

U osenčenim poljima se nalaze podaci (recimo tipa **Atom**). Lista se grana koristeći svojstvo da element lista može biti druga lista odnosno pointer na listu Ovo svojstvo lista do sada nije korišćeno niti sa nalazilo u algoritmima iz prostog razloga što je za predstavljanje skupove takve svojstvo sasvim suvišno i samo bi nepotrebno zakomplikovalo popunjavanja rupa Na slici 2 vidite stablo koje je , napravljeno od lista sa slika 1

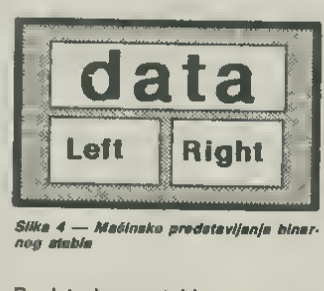
Stablo se dakle može shvatiti i kao razgranata lista što se u jezicima veštačka inteligencija često i forthu obilato koristi



Slika 2 — Stablo napravljeno od liste sa slika 1



Slika 3 — Struktura binarnog stabla



Slika 4 — Mašinsko predstavljanja binarnog stabla

Predstavljanje stabla

Iako se stablo može realizovati preko lista, lako se uočava da se tako gubi prilična količina memorija na nema pointera koji povezuju mala lista koja onda izgravaju čvorove stabla iz grada liste i stabla na slikama 1 i 2 vidi se da čvor stabla traba da sadrži podatak (recimo tipa **Atom**) i konačan broj pointera (linkova) na podstabla Ako je nako podstablo prazno pointer i dalja postoji ali je jednak NIL kao i u slučaju prazna lista Prirodno mašinsko predstavljanja stabla sastoji se dakle od čvorova čija je struktura data na slici 3

Jasno je da broj linkova (tj grana koja izlaze iz svakog čvora) mora biti tikalran u vreme pisanja programa s svakako pre njegovog kompajliranja (u toku izvršavanja programa na može se menjati broj linkova) To, međutim nija nako velika ograničenja, jer se gotovo uvek koristi stablo sa samo dve linka po čvoru Takvo je stablo poznato pod nazivom

Binarno stablo

Čiju definiciju možete dobiti ako u opštoj definiciji stabla umsato sa konačnim brojem pridruženih stavite **sa dva pridružena** Praciznu matematičku definicju možete kao i za stablo naći u RA 30/55

Prirodno mašinsko predstavljanja binarnog stabla je izvedeno iz predstavljanja opšteg stabla Jedina je razlika u tome što nema potrabe za numerisanjam linkova već sa oni jednostavno zovu **left** i **right.**

U tom smislu se govori i o levom i desnom podstablu Čitava teminologija rada se stablima zasnovana je na pojmu levog i desnog podstabla i na pojmovima rodoslovnih veza Rodoslovna stabla su istorijski gledano bila prva praktična primena na samo matematička struktura **stabla** već i gratova uopšte

Ko je ko na stablu

Praciznu matematičku definiciju termina koji se koriste u radu se stablima naći ćete u RA 30/55 pa će ovde biti navedana kroz primer jednog binarnog stabla

Čvor **A** je **koren** (root) stabla a ujedno i **otac** čvorova **B** i **C**, koji su njegovi **sinovi**. I **B** i **C** imaju po dva sina To su **D** i **E** (za **B**) odnosno **F** i **G** (za **C**) Čvor **D** ima samo **desnog sina** (right son) **H** i taj sin je **list** (leaf) odnosno **završni čvor** (**terminal node**) (ovde se iz razumljivih razloga na koristi medicinski termin za sina koji nema , naslednika)

Svi čvorovi stabla (osim **A**) su **potomci** (**descendant**) čvora **A** koji je svima njima **predak** (**ancestor**) Čvor **B** je predak čvorovima **D,E,N** i **I**

jer postoji **put** (path) od S do svakog od njih Iz istog razloga S nije predak ostalim čvorovima (A,C,F,G,J,K)

Stablo je usmeran graf što će reći da grana koja povezuje recimo **C** i **G** znači da postoji put od **C** do **G**, ali ne postoji put od **G** do **C** Ako zamislite kao plan nekog grada (ada su sve ulice (grane) jednosmerne To se mnogo bolje vidi ako se pred očima ima mašinsko predstavljanje stabla.

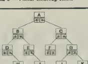
Za svaki čvor se definiše **dubina** (depth) ili **nivo** (level) kao **dužina puta** od korena do datog čvora Pri tome se smatra da je svaka grana dužine 1 Tako je nivo čvora S jedan a nivo čvorova **H,I,J** i **K** tri Vidi se da svi čvorovi koji su u jednom sloju stabla imaju istu dubinu ili nivo

Visina čvora se pak definiše kao dužina najdužeg puta od zadatog čvora do nekog lista. Tako je visina čvora **C** dva iako se od njega do lista može stići i u jednom koraku U programiranju se često dužina puta od jednog do drugog čvora izražava brojem koraka tj brojem programskih ciklusa potrebnih da se taj put „preveli" Visina stabla je visina njenog korena Visina stabla na slikama 5 i 6 je tri jer je tolika visina korena **A**

Ovde iznesena terminologija je praktično jedinstvena u čitavom računarskom svetu Retki feminizirani autori pokušavali su sa terminima „majka" i „ćerka" ali su ostali u manjini Povremeno se sreću i pokušaji uvođenja neutralnih termina „dete" i „roditelj" ali se veoma malo izgleda da ikada steknu popularnost termina „otac" i „sin" koji se nalaze u svim iole značajnijim knjigama kompjuterskih nauka



Slika 5 — Primer binarnog stabla



Slika 6 — Mašinsko predstavljanje stabla

Upotreba stabla

Stabla se koriste za najraznovrsnija poslova u kompjuterskim naukama — od običnog sortiranja pa do stvaranja baze znanja

Upravo to svojstvo stabla da zahteva podatke koji se mogu upoređivati ograničava njegovu primenu pri radu sa skupovima U jednostavnom slučaju koji će u ovoj seriji biti obrađen kad se



Neki istraživači koriste stabla za predstavljanja grafova Pokazano je naime da se svaki graf može predstaviti stablom Isto tako je značajno svojstvo da se svako stablo može predstaviti binarnim stablom što će reći da se svi poslovi mogu raditi isključivo binarnim stablom Naravno da se tako gubi efikasnost korišćenja prostora (kao kad bi se sve radilo sa listama) ali se zato znatno skraćuje programerski posao pošto nije nužno praviti nove programe svaki put kad se pojavi stablo sa granom po čvoru više

Sortiranje i pretraživanje predstavljaju sasvim obična područja primene stabala Pri tom se mogu koristiti praktično isti algoritmi za oba posla Stablo efikasno čuva samo one podatke među kojima postoji neka relacija poretka Pri samom razpoređivanju elemenata po stablu vrši se praktično i sortiranje Potrebno je samo „proći" stablom po nekom od jednostavnih postupaka (preorder inorder, postorder) i sortirani podaci su na raspolaganju

Atom sastoji samo od stringa ili broja poređenje se lako definiše Šta međutim raditi ako **Atom** može da bude i matrica ili složeno polje podataka?

Postoje srećom i područja primene u kojima se uslov uporedljivosti vrlo lako zadovoljava Tada stabla pokazuju svu svoju moć i višestruko ubrzavaju rad

Jedan od klasičnih primera je konstrukcija tabele simbola što je potrebno svakom kompajleru i asembleru Budući da su simboli stringovi to je stablo gotovo idealna struktura za ovaj posao Ovde se međutim mogu pojaviti neke vrlo nezgodne situacije Ukoliko su simboli na ulazu već u nekom poretku stablo će degenerisati u listu Čak i kad nisu svi podaci uređeni svaka iole veća grupa uređenih podataka može ozbiljno da smanji brzinu manipulisanja

Sledeća primena je sintaksna analiza izraza Pošto se sintaksa izraza zadaje nekim skupom pravila to ova pravila zadaju „poredak" elemenata konkretnog izraza i omogućavaju njegovo raščlanjivanje i razvijanje po stablu Nad tako razvijanim izrazom mogu se efikasno izvršavati razne operacije Ovde se ne radi više o nalaženju pojedinog elementa u zavisnosti od nekog poretka već o identifikovanju funkcionalnih grupa nekog (programskog) jezika i razmeštanju takvih grupa radi što odljeg i efikasnijeg izvršavanja Operacije sa funkcionalnim grupama su izuzetno jednostavne u terminima stabla budući da svako podstablo se svojom hijerarhijskom strukturom (otac — sinovi) može predstavljati neku grupu operacija.

Zbog ovih svojstava stablo se koristi i u programima za raspoznavanje prirodnog jezika gde je složenost posla toliko velika da bi bez stabla bilo gotovo nezamislivo i pokušati bilo šta urediti

Simbolička manipulacija algebarskim izrazima se takođe oslanja na sposobnost stabla da efikasno održava informacije o funkcionalnim grupama Ovog puta radi se o grupama matematičkih operacija i funkcija. Još je daleke 1967 godine Donald Knuth (čita se Donald Nut a ne Knut) u svojoj danas već klasičnoj knjizi The Art of Computer Programming objavio kompletan algoritam za diferenciranje i delimično sređivanje algebarskih izraza Algoritam je naravno zasnovan na stablu po koma je razvijen izraz koji treba da bude diferenciran Hewlett-Packardov kalkulator HP—28C kog su čitaoci „Računara" već imali priliku da upoznaju ima softver pisan na forthu samo zato što je njima izuzetno lako organizovali rad sa stablima i razvijanje izraza po stablu Osim automatskog diferenciranja i sređivanja ovaj kalkulator ima komande za manuelnu obradu izraza Te komande ne rade ništa drugo osim što menjaju pointere u čvorovima stabla

Binarno stablo i skupovi

Glavna prednost binarnog stabla nad listom je efikasnost pretraživanja Za binarno stablo sa **N** elemenata potrebno je u proseku **lgN** elementarnih operacija (poređenje + podešavanje pointera) za pronalaženje zadatog elementa dok je za listu sa **N** elemenata potrebno **N/2** operacija (u srednjem) Već za svega 32 elementa rad sa stablom je oko 3 puta brži za 1024 elementa je čak 100 puta brži itd

Kada se skup predstavlja stablom nad njim se mogu izvršiti operacije **MEMBER INSERT, DELETE, UNION, SPLIT** i **MIN** Dok su operacije **SPLIT** i **MIN** izuzetno jednostavne zbog same strukture stabla koja pored skupa reprezentuje i relaciju poretka koja je za taj skup zadata. Operacije **MEMBER** i **INSERT** su takođe sasvim jednostavne dok je **DELETE** tvrd orah Uklanjanjem jednog elementa stabla potrebno je preurediti skoro čitavo stablo kako bi se zadržao poredak

1 — levi sin je manji od oca
2 — desni sin je veći od oca

Na osnovu ovakvog poretka zna se da su svi elementi levog podstabla nekog čvora veći od svih elemenata desnog podstabla istog čvora. Ukoliko se pri unošenju podataka daju podaci u rastućem nizu oni će obrazovati listu u desnom podstablu a ako se daju u opadajućem nizu obrazovaće listu u levom podstablu

Da bi se sprečila ovakva degeneracija uvede se posebni algoritmi balansiranja stabla koji „paze" da čvorovi budu uvek ravnomerno raspoređeni u obe podstabla svakog čvora Dva najpoznatija algoritma su AVL algoritam i algoritam za 2—3 stablo Ovi algoritmi na žalost unose velika komplikacije u rad i produžavaju izvršavanja jedne osnovne operacije tako da se u nekim primenama postavlja pitanje isplati li se uopšte koristiti balansirana stabla

U sledećem nastavku ćemo videti kako izgleda konkretna realizacija skupa kao stabla Biće naravno, korišćen i kompletan algoritam popunjavanja rupa koji je razvijen u prošlom nastavku

Žarko Berberski

Osnovne računske radnje

Mikroprocesor Z80, kao i većina njemu sličnih, jedino raspoznaje i izvršava operacije celobrojnog sabiranja i oduzimanja. Sva ostalo mora da uradi programar, počev od množenja i deljenja celih brojeva, pa do operacija sa racionalnim brojevima i dalje do elementarnih matematičkih funkcija. Pisanje takvih programa nije nikakav bauk, ali se dosta stvari mora znati. Cilj ovog teksta je da pomogne početnicima u razumevanju i pisanju programa za operacije nad racionalnim brojevima



Za primer programa koji obavljaju osnovne matematičke radnje uzeli smo odgovarajuće spektru mova rutine i izmenili ih donekle da bi bile jasnije kao i da bismo uklonili neke bagove. Za razumevanje ovih programa dovoljno je poznavati osnova mašinskog programiranja i neke stvari o kojima smo već u više navrata pisali ali ćemo ih i sada ukratko ponoviti.

Racionalan broj je u memoriji računara zapisan u tzv. *formi pokretnog zareza*. Zapis zauzima pet bajtova od čega prvi označava eksponent uvećan za 128 a preostala četiri čine mantisu tačnije bitove iza decimalne tačke. Mantisa je normalizovana tako da joj je prvi bit uvek jedinica što znači da je vrednost mantise između 0.5 i 1. Taj prvi bit mantise koji se podrazumeva ima ulogu predznaka broja (jedinica za negativne brojeve).

Prvo ćemo razmotriti množenje i deljenje jer su umnogome slične operacije. O sabiranju i oduzimanju ćemo govoriti sledeći put.

Množenje

Dva broja $X2^a$ i $Y2^b$ množe se tako što im se pomnože mantise X i Y a eksponenti saberu: $a+b$. Rezultat ima vrednost $XY2^{a+b}$. Rezultujuće mantisa može imati narušenu normalizaciju udesno tako da prvi bit nije jedinica. Međutim tada drugi bit mora biti jedinica i normalizacija se postiže pomeranjem ulevo za jedno mesto pri čemu eksponent treba umanjiti za jedinicu.

Deljenje

Dva broja $X2^a$ i $Y2^b$ dele se tako što im se podele mantise a eksponenti oduzmu: $X/Y2^{a-b}$. Normalizacija može biti narušena ali se uspostavlja odgovarajućim šiftovanjem mantise.

Programi koje objašnjavamo zahtevaju na ulazu adrese argumenata u HL i DE. Na izlazu je rezultat smešten na mestu prvog argumenta i HL pokazuje na njega.

Potprogram FETCH uzima iz memorije dve broja nad kojima će se izvršiti neka od računskih operacija i smešta ih u registre mikroprocesora. Na ulazu HL pokazuje na eksponent prvog broja a DE na eksponent drugog broja. Na izlazu su AF i HL očuvani prvi broj je prenet u H B C C B a drugi u L D E D E. Ulazni sadržaj akumulatora je ujedno smešten u drugi bajt prvog broja kao indikator predznaka rezultata. Međutim to je od značaja samo za množenje i deljenje.

Potprogram PREP obavlja pripremu broja za množenje ili deljenje. Ta se priprema uglavnom sastoji u setovanju prvog bita mantise jer je jedinica njegova prava vrednost. Predznak broja se kombinuje sa ulaznim sadržajem akumulatora što omogućuje formiranje znaka rezultata ukoliko se PREP pozove najpre za jedan broj a zatim za drugi.

Potprogram INCR uvećava za jedinicu mantisu rezultata H L H L. Poziva se prilikom zaokruživanja mantise pri množenju i deljenju. Ukoliko pri dodavanju jedinice dođe do prekoračenja biće uvećan za jedinicu sadržaj registra B koji igra ulogu popravke eksponenta rezultata.

Program MULTPL obavlja množenje dva broja u pokretnom zarezu. Na ulazu HL pokazuje eksponent prvog broja a DE eksponent drugog broja. Obično se oba broja nalaze na aritmetičkom steku jedan iznad drugog ali slobodno mogu biti bilo gde u memoriji. Samo je potrebno zadati njihove adrese. Na izlazu adrese rezultata je u HL kao na ulazu. Prvi deo programa sastoji se u pripremi i prikupljanju brojeva. Ukoliko je prvi broj nula povratak se vrši neposredno jer je i rezultat nula. Ukoliko je drugi broj nula izlaz se vrši preko Z-RSLT gde se nula dovodi na mesto rezultata.

Glavna petlja za množenje mantisa. Prvi broj je u D E D E a drugi u B C C A. Rezultat se formira u H L H L postupnim šiftovanjem udesno. Šiftovanje se deli prenosi na B C C A, na mesto istisnutih bitova.

Glavni posao oko množenja je obavljen. Preostaje da se mantise normalizuje i saberu eksponenti. Na izlasku iz petlje M-LOOP rezultat je u H L H L. C i B. Sveke manipulacije se mantisom tj. pomeranje ulevo ili udesno pri normalizaciji podrazumeva i korekciju eksponenta: uvećanja pri pomeranju udesno i umanjenje pri pomeranju ulevo. Registar B koristiće se za formiranje korekcije eksponenta sa sledećim značenjem: ako je B=0 eksponenti treba umanjiti za jedinicu; ako je B=1 eksponenti ne treba korigovati; ako je B=2 eksponenti treba uvećati za jedinicu. Početna vrednost biće B=1.

Zaokruživanje normalizovane mantise vrši se ukoliko je 33-i bit jedinica. Ovaj bit je posle normalizacije smešten u indikatoru prenosa.

Eksponent rezultata dobija se prostim sabiranjem eksponenta poslednjih brojeva uz dodavanje korekcije formirane u registru B. Treba samo obratiti pažnju da je korekcija B za jedan veća od stvarne da bi ostale ove vreme pozitivne. To znači da od konačnog eksponenta treba oduzati jedinicu.

```
FETCH   PUSH  HL      Sačuvaj adresu prvog broja.
        PUSH  AF      Sačuvaj status
        LD    C,(HL)  Prvi bajt prvog broja
        INC   HL
        LD    B,(HL)  Drugi bajt.
        LD    (HL),A  Predznak rezultata
        INC   HL
        LD    A,C
        LD    C,(HL)  Treći bajt
        PUSH  BC      Sačuvaj drugi i treći bajt.
        INC   HL
        LD    C,(HL)  Četvrti bajt
```

```
        INC   HL
        LD    B, (HL)       Peti bajt.
        EX    DE, HL        Adresa drugog broja ide u HL.
        LD    D A           Eksponent prvog broja.
        LD    E, (HL)       Eksponent drugog broja.
        PUSH  DE            Sačuvaj eksponente.
        INC   HL
        LD    D (HL)        Drugi bajt drugog broja.
        INC   HL
        LD    E, (HL)       Treći bajt.
        PUSH  DE            Sačuvaj drugi i treći bajt
        EXX                 Alternativni set registara
        POP   DE            Drugi i treći bajt drugog broja.
        POP   HL            Eksponenti oba broja
        POP   BC            Drugi i treći bajt prvog broja
        EXX                 Nazad u glavni set.
        INC   HL
        LD    D (HL)        Četvrti bajt drugog broja
        INC   HL
        LD    E, (HL)       Peti bajt.
        POP   AF            Obnovi status
        POP   HL            Obnovi adresu prvog broja
        RET                 Povratak
PREP    LD    B A           Sačuvaj predznak iz prethodnog poziva
        LD    A, (HL)       Uzmi eksponent broja
        AND   A             Da li je nula?
        LD    A, B          Obnovi stari predznak
        RET   Z             Povratak ako je broj nula. Indikator preneса je
                            resetovan.
        INC   HL            Pomeri se na prvi bajt mantise
        XOR   (HL)          Formiraj predznak rezultata
        SET   7, HL         Prava vrednost mantise
        DEC   HL            Vrati se na eksponent.
        SCF                 Signal broj nije nula.
        RET                 Povratak.
INCR    INC   L             Uvećaj četvrti bajt.
        RET   NZ            Povratak ako nema prekoračenja.
        INC   H             Uvećaj treći bajt
        RET   NZ            Povratak ako nema prekoračenja
        EXX
        INC   L             Uvećaj drugi bajt.
        EXX
        RET   NZ            Povratak ako nema prekoračenja.
        EXX
        INC   H             Uvećaj prvi bajt.
        EXX
        RET   NZ            Povratak ako nema prekoračenja
        EXX
        LD    H #80         Pomeri mantisu udesno za jedan bit.
        EXX
        INC   B             Uvećaj korekciju eksponenta za jedan.
        RET                 Povratak
MULTPL  XOR   A             Priprema za prvi poziv PREP
        CALL  PREP          Priprema prvog broja. Njegov predznak bi_ s
                            vraćen u registru A.
        RET   NC            Povratak ako je prvi broj nula
        EX    DE, HL        Adresa drugog broja u HL.
        CALL  PREP          Priprema drugog broja
        EX    DE HL         Stari redosled adresa
        JR    NC Z_RSLT     Skok ako je drugi broj nula
        CALL  FETCH         Uzimanje brojeva iz memorija
        XOR   A             Brisanje indikatora preneса.
        EXX
        PUSH  HL            Sačuvaj eksponente H I L
        SBC   HL, HL        Brisanje H L' — priprema rezultata.
        EXX
        PUSH  HL            Sačuvaj adresu rezultata.
        SBC   HL, HL        Početna vrednost rezultata u H L HL je 0
        LD    A, B          Poslednji bajt prvog broja.
        LD    B 32          Ulaz u petlju za množenje 32 bita.
M_LOOP  EXX
        RR    B             Mantisa drugog broja B C CA pomera
        RR    C             se udesno za jedan bit, čime se u
        EXX                 indikator preneса dovodi novi bit za
        RR    C             množenje. Na mesto upražnjenog bita
        RRA                 slave strane dolazi novi bit rezultata
        JR    NC, NO_ADD    Skok ako nema množenja
        ADD   HL, DE        Mantisa D'E'DE dodaje se rezultatu
        EXX                 u H L HL.
        ADC   HL, DE
        EXX
NO_ADD  EXX                 Rezultujuća mantisa
        RR    H             u H L HL pomera se za jedan bit
        RR    L             udesno da bi se obezbedilo pravilno
        EXX                 , potpisivanje pri sledećem
        RR    H             sabiranju
        RR    L
        DJNZ  M_LOOP        Ponovi 32 puta za svaki bit.
        INC   B             Korekcija eksponenta jedinica.
        EXX
        BIT   7,H           Da li je mantisa već normalizovana
        EXX
        JR    NZ, M_RND     Ako jeste idi napred.
        DEC   B             Korekcija eksponenta. nula.
        ADC   HL. HL        Pomeranje ulevo
        EXX
        ADC   HL,HL
        RL    B             34-I bit ide u indikator preneса.
M_RND   CALL  C INCR        Zaokruži mantisu ako treba.
        PUSH  HL            Prenesi poslednje dve bajta mantise
        EXX                 rezultata iz HL u D E_ taka da se
        POP   DE            rezultat sada nalazi u H L D E
        POP   BC            Uzmi se steka adresu rezultata
        PUSH  BC            ali je ondah I vrati
        INC   BC            Pomeri se na prvi bajt mantive
        LD    A,(BC)        Uzmi predznak rezultata I
        RL    H             dovedi ga na svoje mesto u
        RLA                 registru H
        RR    H
        EXX
        XOR   A             Jedna nula će biti potrebna kasnije
        POP   HL            Skini se steka adresu rezultata
        EX    (SP),HL       Donesi u HL eksponente a adresu vrad
        LD    E H           Eksponent prvog broja.
        LD    C B           Korekcija eksponenta rezultata
        LD    H,A           Viši bajtovi registara HL. DE I BC se
        LD    D,A           postavljaju na nulu. Sada je u HL prvi
        LD    B,A           eksponent, u DE drugi a u BC korekcija.
        ADD   HL, DE        Sabiranje eksponenata
        ADD   HL, BC        Dodavanje korekcija
        DEC   HL            Zbog uvećana korekcija
        DEC   H             Oduzimanje 256 od eksponenta rezultata
                            jer su polazni eksponenti bili uvećani
                            za po 128
        LD    A,L           Prenesi eksponent u akumulator
        INC   H             Da li je eksponent negativan_
        POP   HL            Uzmi adresu rezultata.
MID_END JR    NZ,E_POS      Skok ako je eksponent pozitivan.
        CP    #81           Da li je eksponent ispod −128
        JR    NC,E_OK       Skok napred ako nije.
Z_RSLT  XOR   A             Memorija će se popuniti nulama.
        LD    B,5           Zapis broja ima pet bajtova.
FILL_Z  LD                  (HL),A Unesti nulu u memoriju.
        INC   HL            Pomeri se na sledeći bajt.
        DJNZ  FILL_Z        Ponovi petlju pet puta.
        LD    BC −5         Priprema za obnavljanje adrese
        ADD   HL, BC        Vrati se na početak broja
        RET                 Povratak iz programa.
E_POS   CP    #80           Da li je eksponent +128 I više
        JP    NC OVERFL     Ako jeste to je prekoračenje
E_OK    ADD   A,#80         Uvećaj eksponent za 128
        LD    (HL), A       Smesti eksponent u memoriju.
        INC   L             Pomeri se na sledeći bajt.
        EXX
        PUSH  DE            Mantisu iz H L D'E prenesi na
        PUSH  HL            mašinski stek.
        EXX
        POP   BC            Uzimaj se steka
        LD    (HL), B       redom bajtove mantise I
        INC   HL            Smeštaj ih u memoriju
        LD    (HL) C
        INC   HL
        POP   BC
        LD    (HL) B
        INC   HL
        LD    (HL) C
        DEC   HL
        DEC   HL
        DEC   HL
        DEC   HL
        RET                 Povratak iz programa
DIVISN  EX    DE.HL         Adrese drugog broja u HL.
        XOR   A             Pripreme se prvi poziv PREP
        CALL  PREP          Priprema drugog broja i uzimanje njegovog
                            predznaka.
        JP    NC OVERFL     Skok ako je drugi broj nula
        EX    DE, HL        Vraćanje u HL adrese prvog broja
        CALL  PREP          Priprema prvog broja i formiranje predznaka
                            rezultata.
        RET   NC            Povratak ako je prvi broj nula
        CALL  FETCH         Prikupi brojeve iz memorije
        EXX
        PUSH  HL            Sačuvaj eksponente a B C prenesi
        LD    L, C          u H L
        LD    H B
        EXX
```

```
        PUSH  HL        Sačuvaj adresu rezultata e D I C
        LD    L, B      prenesi u L I H  Sada je mantisa
        LD    H C       prvog broja u H L HL.
        XOR   A         Resetuj indikator prenosa za prvi
                        ulaz u petlju za deljenje
        LD    B −33     Početna vrednost brojača
        JR    D_STRT    Skok u petlju
D_LOOP  RLA             Rezultat se formira u B C CA pomera-
        RL    C         njem ulevo i uzimanjem novog bita
        EXX             iz indikatora prenosa
        RL    C
        RL    B
        EXX
SHIFT_X ADD   HL, HL    Pomeri deljenik ulevo pre sledećeg
        EXX             oduzimanje delioca
        ADC   HL, HL
        EXX
        JR    C S_ONLY  U slučaju prekoračenje skoči napred i obavi
                        prosto oduzimanje
D_STRT  SBC   HL DE     Oduzmi delilac od deljenika
        EXX
        SBC   HL, DE
        EXX
        JR    NC N_RSTR Skok ako je oduzimanje bilo uspešno
        ADD   HL, DE    Obnovi sadržaj deljenika.
        EXX
        ADC   HL, DE
        EXX
        AND   A         Resetuj indikator prenosa.
        JR    CT_ONE    Skok napred
S_ONLY  AND   A         Priprema za oduzimanje
        SBC   HL DE     Oduzmi delilac od deljenika
        EXX
        SBC   HL, DE
        EXX
N_RSTR  SCF             Novi bit rezultata
CT_ONE          IHC     Uvećaj brojač
        JD      M D_LOOP  Hazad u petlju sve dok je brojač negativan
                        −32 DO −1
        PUSH  AF        Novi bit na stek.
        JR    Z SHFT_X  Nazad u petlju
        LD    L, A      Prenesi A I C u L I H
        LD    H, C
        POP   AF        Uzmi bit 34 i prenesi ga u
        RR    E         registar E.
        POP   AF        Bit 33 prenesi u indikator prenosa
                        Rezultat je sada u B C HL c E.
        EXX
        LD    L C       Prenesi B C u H L
        LD    H B
        BIT   7 H       Da li je mantisa normalizovana
        EXX
        JR    NZ D_RND  Skok ako je mantisa već normalizovana.
        DEC   B         Korekcija eksponenata
        ADC   HL, HL    Normalizacija ulevo
        EXX
        ADC   HL, HL
        EXX
        RL    E         Bit za zaokrušivanje
D_RND   CALL  C INCR    Zaokrušivanje eko treba.
        PUSH  HL        Prenesi mantisu iz H L HI u
        EXX             H L D E
        POPE
        POP   BC        Adresa rezultata
        PUSH  BC
        INC   BC        Pomeri za na predznak broja
        LD    A, (BC)   Formiraj predznak rezultata
        RL    H
        RLA
        RR    H
        EXX
        XOR   A         Nula za kasnije
        POP   HL        Adresa rezultata
        EX    (SP) HL   Uzmi eksponente polaznih brojeva
                        a vrati adresu na stek.
        LD    e I       Drugi eksponent u E.
        LD    I h       Prvi eksponent u L
        LD    c b       Korekcija eksponenta u C
        LD    h e       Nula se dovodi u H D I B
        LD    d e
        LD    b e
        SBC   hl de     Oduzimanje eksponenata
        ADD   hl bc     Korekcija eksponenta.
        LD    e I       Eksponent u A.
        INC   H         Da li je eksponent negativan
        POP   HL        Uzmi adresu rezultata
        JP    MD_END    Skok na zajednički kraj množenja i deljenje
```

Delje sledi deo koji je zajednički za množenje i deljenje U HL je adrese rezultate eksponent rezultate je u akumulatoru mantisa se odgoverejućim predznakom u H L D E a indikator nule čuve informaciju o predznaku eksponenta nula ako je negativen

Eksponent je negativen i suviše mali ispod −128 tako da će rezultat tehtički biti nule

Pozitiven eksponent ne sme biti veći od 128 uključujući i tu vrednost Sveke vrednosi iznad +127 proglaševe se za prekoračenje I vrši se skok na neku adresu OVERFL gde bi trebalo de se nalazi rutine za prijavu greške i prekid programe

Eksponent je u dozvoljenim granicame (između −128 I +127 Preostaje de mu se dode 128 I de se ceo rezultat smesti na svoje mesto u memoriji

Program DIVISN obavlja deljenje dve broje u pokretnom zarezu Ne ulazu HL pokazuje eksponent prvog broja e DE eksponenti drugog broja Brojevi se mogu nalaziti bilo gde u memoriji sli je uobičajeno da se smešteju jedan iznad drugog ne aritmetičkom steku Na izlazu adrese rezultata će biti u HL e to je pozicija prvog ulaznog broja

Prve naredbe programe obavljaju pripremu i prikupljanje brojeva iz memorije Ukoliko je drugi broj nula bez obzira na vrednost prvog vrši se skok na OVERFL da bi se prijavila greške o prekoračenju Ukoliko je potrebno razlikovati ovakvo prekoračenje od uobičajenog može se skok vršiti na nekakvu drugu adresu gde će se prijaviti greška deljenje nulom

Glevne petlje za deljenje mentise Deljenik prvi broj je u H L HL e delilac drugi broj je u D E DE Rezultat se formira u B C CA šiftovanjem ulevo za prvih 32 bitova mantise pri čemu brojač B uzima vrednosti od −32 do −1 Bitovi 33 i 34 koji su potrebni za normalizaciju i zaokrušivanje mentisa smeštaju es na mašinski slek za vrednosti brojača 0 I 1

Deljenje je obavljeno šiftovi mentise rezultate su u B C CA i na mešinskom sleku Postupak normalizacija i zaokrušivanja biće sličan kao kod množenje Pre svega rezultat se prenosi u H L HL Bit 33 ide u indikator prenosa a bit 34 u registar E Korekcije eksponente se kao kod množenje formira u registru B s tom razlikom što u slučeju deljenja početna vrednosi korekcije iznosi nule

Jovan Skuljan

GALAKSIJA

u avguatovakom broju

Nanotehnologija

STAPANJE ČOVEKA I MAŠINE

Ekskluzivni intervju se Stjuartom Hamerofom vodećim emerličkim stručnjakom za mešine veličine molekula i stepenje bio i informatičke tehnologije

Naurokompjutsri

MOZAK I NJEGOV DVOJNIK

Veštečke neuroneke mreže od kompjutera stvaraju mašinu koje se u neklm elementime takmiči sa ljudskim mozgom

Laaari i kompjutari
RAT BUDUĆNOSTI

Intsligsntni saobraćaj
SUPERVOZ

Putovanja na Mars
LET BEZ MOTORA

Ekskluzivno
BAJKONUR BEZ TAJNI

Radix List Sort

Brže od najbržeg

Sa sortiranjem reči sa veoma često sraćem u praksi primenjujući najrazličitija algoritma, ali rešenjima nikada nisam bio zadovoljan do kraja. U trenutku programerskog proavatijanja, pronašao sam potpuno novi algoritam za sortiranja koji prevazilazi ava postojeće. Radoat je, na žalost, bila kratkog vaka — pažljivim pregledom literature ustanovio sam da su osnove algoritma prvi put opisane 1929 godina, a primenjivan je još na mahaničkim mašinama za sortiranje!!

Baz obzira na ovu anegdotu ostaja činjanica da se radi o pomalo zaboravljanom algoritmu sa zaista izvanrednim karakteristikama koji je posebno pogodan za današnji tip računara. To je jedini poznati algoritam čija je vreme sortiranja linearno zavisno od broja elemenata koji se sortiraju potpuno je nezavisan od početnog rsaporede elemanata kretko i lako aa kodira Ali da kranemo redom

Dva prolaza

Osnovna ideja RADIX LIST SORT algoritma je veoma jednostavna

Pretpostavimo de treba sortirati niz od 10 jednocifranih brojeva koji mogu biti samo 0 1 ill 2

1 0 0 2 1 2 9 2 2 0

Prebrojaćemo pojavljivanja avake od cifara u ovom nizu

0 ima 4
1 ima 2
2 ima 4

Poznajući ove vrednoati lako je zaključiti gda će se u sortiranom nizu naći avaka od cifara

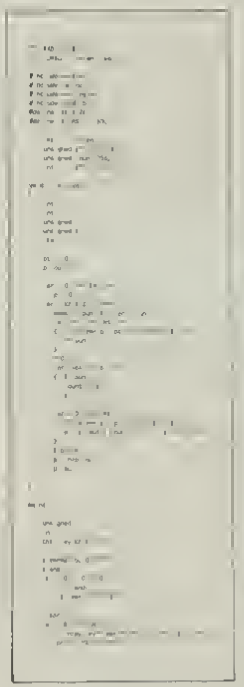
0 će zauzimati meste od 1 do 4
1 će zauzimati mseta od 5 do 6
2 će zauzimati mseta od 7 do 10

Dakle možemo uzimeti jednu po jednu oltru iz niza i prabacivati je u novi niz na sledeći način

1 ide na mesto 5 sledaće jedinica koja sa pojavi Ida na mseto 6
9 Ida na mesto 1 sledaća nula će ići na mesto 2.
0 Ida na mesto 2 sledeća nula će ići na mesto 3
2 Ida na mseto 7 sledeća dvojka će ići na mseto 6 itd

Rezultujući niz je nakon ovog prolaza sortiran

Algoritam zahteva dva prolaza dva memorijska područja za elemente koji se sortiraju i jedan niz koji služi kao brpjač Uopštavanja na sortiranja višecifrenih brojeva (višeslovnih reči svejadno) zahteva da se čitav postupak ponovi onoliko puta koliko citara sedrže brojevi koji se sortiraju primanjujući postupak od kraja ka početku — prvi prolaz sortira u odnosu na poslednju cifru sledači na pretposlednju i tako redom do nejznačejnija citra



i dva nedostatka

Iz ovoga sa odmah mogu uočiti dva glavna nedostatka ovog algoritma Ako se sortiraju jako dugački ključevi program će tražiti mnogo vremana ne sortiranja poslednjih citara koje najčešće malo utiču na konačan rsaporad Drugi nedostatak ja utrošak memorija jer ja potreban dosatni prostor za prematšanja čitavog niza Ovaj poslednji nedostatak ja i glevni razlog zašto je algoritam svojavremeno zaboravljan Danse ja jsano da se problam lako može rešiti upotrabom pointera a a obzirom da se nizu uvek pristupa sekvencijalno lako se mogu koriatiti i tehnika ulančenih lista čima se memorija optimalno iskorišćava u slučaju sortiranja ključava najednake dužine

Za algoritam je karakteristično da zapravo uopšte ne koriati poređenja već samo indaksiranja Ujedno jedini akokovi au oni kojima se ostvaruja patlje tako de je algoritam vsoma pogodan za mikroprocesore kao što je familija 80xxx kod kojih sa instrukcije izvršavaju iz reda koji sa „unapred formira Na alici je dat program koji smo koristili za testiranje algoritma Dugačak jednodimanzioni niz je popunjan slučajnim karakterima iz AS-CII sata od 64 do 95 Niz ja zatim tretiran različito — kao niz od 16000 slogova dužina 4 zneke kao niz od 12000 slogove dužine 5 znakova i tako redom Nad svakim od njih ja primenjan naizmenična RADIX LIST algoritam i QUICK sort iz bibliotaka funkcija TURBQ C-a Rezultati su izmereni u sekundame na AT računaru (10MHz) i pružaju veoma interesantnu sliku

	QS	RLS
16000/4	15 60	3 00
12000/5	14 40	3 00
10000/6	9 40	2 50
6000/10	8 10	2 40
3000/20	3 50	2 50

Zajedno su jači

Promana dužina sloga nija znatno uticala na vremana ostvarena sa RLS jer ja broj slogove moreo biti amanjan (ah ta granica od 64K!)



Da amo nešto iskompilkovali program i proverili aituaciju sa 12000 slogova dužine 20 znakova jasno bi sa uočilo porast vremana zbog povećana dužine sloga (vrema bi približno bilo jednako sa QS) Međutim QS možemo da stavimo na daleko veća muka — eko mu prosladimo već sortiran niz (najgori slučaj za ovej algoritam), vreme će drastično porsati (probali smo verziju 10000/6 QS sa mučio 104 sekunda!) dok ja RLS potpuno neosetijiv na ovakva aituacija

Pri realizacji algoritma moguća au različita rešenja u zavisnoali od konkretna situacije interesantno je de aa algoritam može dosta atikano primenti i na sortiranje velikih datoteke sa diaka S obzirom da se prabrojevenje cifara u svim kolonama može obavlti u jednom sakvencijalnom čitenju ostaja da se reš prabacivanje u novi niz pri sradivanju svaka od kolona, koja zahteve slučajan pristup Problam ja lako rešiv ako nam je na raspolaganju bar nešto slobodna memorija kae privremeni bafer

Posebno je interesatno kombinovenja ovog algoritma sa drugima čima sa može pravazići njegov nedostatak pri sortiranju jako dugačkih ključeva Ponakad ja dovoljno algoritam primanti samo na najznačejniju cifru — nakon ovoga niz je već pničino sortiran pa alikano posao mogu da nastava i algoritmi kao što je BUBBLE sort

Zoran Životić

Sortiranje izraza/„spektrum"

Brzi Gonzales

Bajzik „spektruma" i drugih kućnih računara ne može se pohvaliti brzim sortiranjem stringova, baz obzira na to kakav je algoritam primanjan i koliko je programar vašt. S mašincem ja, naravno, drukčija. Ovaj program sortira po rastućem redosledu ato imena za manja od dva eekunda. Relokatibilan je, ime evega 90 bajtova, na treba mu dodatna mamorija za radni prostor, pa maksimalna dužina datoteka koja se sortira praktično zeviei samo od raspoložive mamorija računara.

Izrazi koji se sortiraju mogu biti memorisani u sakr zaplau (na primer spisak imena, lista s nazivima proizvoda i slično) ili u apsolutnim brojevima (ako su to celi brojevi od 0 do 255 kao što su brojevi u kombinacijama za loto) Program podjednako uspešno sortira i jednu i drugu vrstu zapisa

Svi izrazi moraju po definiciji imati istu dužinu Ako u stvarnosti nau podjednaki (spisak imena učenika u jednom odeljenju na primer) dužina se daliniše prama najdužem izrazu a ostalak bajtova u kraćim izrazima popunjava se kodom 32 (aski zapis) ili vrednošću koju odredi programar (apsolutni brojevi) izrazi na moraju imati granićnika za početak i kraj — dovoljno je i to što program zna koliko bajtova zauzima svaki od njih — pa se prilikom memorisanja prosto „slažu jedan iza drugog

Jedan izraz na može zauzimati više od 255 bajtova a za dužinu datoiske nema ograničenja — ogem naravno raspoloživa memorije

Da bi program uopšte mogao da radi potrebno je da mu se saopšte in podatka

1 **Adrese datoteke** Ovaj podatak se zapisuje u dva bajta neposredno ispod startne adres programa. Program uma da utvrdi koja su to dva bajta bez obzira da li je relocirari ili ne Ovim rešenjem je posegnuta relokatibilnost i programa i datoteke što širom otvara vrata raznim aplikacijama

Obrazac za dvobajtni zapis broja verovatno znaiš Račimo da sis program učitali na 60000 Pra starta naredbom RANDOMIZE USR 60000 potrebno je da na 59998 upišete INT (ADR/256) a na 59999 ADR-INT (ADR/256) Ukoliko vam je datoteka uvek na istoj adresi dovoljno je da ovo upisivanja obavite samo jednom r da potom program snimila na traku zajedno s ta dva bajta ispred njega U tom slučaju datoteku nećeiis više moći da pomerate bez novog upisivanja adrese ali će pegram i dalja ostali relokatibilan

2. Dubina pretrage Morate odredili koliko će slova (ili brojeva) program uporađivati Nakad je dovoljno da se uporeda samo prva dva znaka na primar prilikom sortiranja imena Petar i Pavla Ali nakad je potrebno i više Da bi se sortirsia imena Milan i Milanka iraba uporedi šest slova Dubina pretrage poređuje se brojem od 1 do 255 Što je taj broj manji program brže radi.

Ova vrednosi se upisuje u prvi bajt ispod datoteka

3. Dužina izraza Već smo rakli da dužina izraza takođe može biti od 1 do 255 Sada dodajamo da se ta vrednost upisuje odmah ispod dubina to jest u drugi bajt ispod datoteke

Pošto su dubina i dužina stalni parametri nećete pogrešiti ako posle prvog upisivanja ovih vrednosti snimite datoteku zajedno s dva bajta ispod nja Tako ćeta izbeći upisivanja prilikom svakog rada a datotekom a njena relokatibilnosi će i dalje biti očuvana kao i mogućnost proširivanja ili skraćivanja

I još jedna napomena podrazumeva se da ja prvi bajt iza datoteke „prazan to jest da je njegova vrednost nula Program tu nulu shvata kao graničnik

Ova uputstva su dovoljna za rukovanja programom Ostatak teksta nije obavezan za čitanje — to je komentar programa namanjan radoznalim hakarima

```
L
00001 ;-----------------------
00002 .   SORT
00003 ;-----------------------
00010         ORG  52000
00020         PUSH BC
00030         POP IX
00040         LD H (IX-1)
00050         LD L (IX-2)
00060         PUSH HL
00070         POP IX
00080 ;-----------------------
00090 LAB1    PUSH IX
00100         POP HL
00110         LD (ADR),HL
00120         LD (IX-3) 0
00130 ;
00140 LAB2    LD HL,(ADR)
00150         LD D H
00160         LD E,L
00170         LD B,0
00180         LD C,(IX-2)
00190         ADD HL BC
00200         LD (ADR) HL
00210         XOR A
00220         OR (HL)
00230         JR NZ,LAB3
00240         XOR A
00250         OR (IX-3)
00260         JR NZ LAB1
00270         RET
00280 ;-----------------------
00290 LAB3    LD B (IX-1)
00300 PT1     LD A,(DE)
00310         CP (HL)
00320         JR C LAB2
00330         JR NZ LAB4
00340         INC DE
00350         INC HL
00360         DJNZ PT1
00370         JR LAB2
00380 ;-----------------------
00390 LAB4    LD HL,(ADR)
00400         LD D-H
00410         LD E,L
00420         LD C (IX-2)
00430         LD B 0
00440         SBC HL BC
00450 PT2     LD A,(DE)
00460         LD B (HL)
00470         LD (HL),A
00480         LD A,B
00490         LD (DE),A
00500         INC DE
00510         INC HL
00520         DEC C
00530         JR NZ,PT2
00540         LD (IX-3),1
00550         JR LAB2
00560 ;-----------------------
00570 ADR     EQU 23720
```

Cena relokatibilnosti

10—70. Utvrđivanje adresa

Da bi program utvrdio na kojoj je adresi datoteka mora najpre da sezna gde ga je korisnik učitao Za to se služi pogodnošću koju pruža operativni sistem „spektruma Na ulazu ragistraski par BC sadrži vrednosi funkcija USR Tu vrednosi prako siaka preuzima IX a HL zaiim čitanjem bajtova IX—1 i IX—2 preuzima adrseu datoteka koju sta vi upisali U daljem radu tu vrednost treba da ima IX pa je seto prako siaka preuzima od HL Ovim peiijanjem na izgled suvišnim, piaćena je relokatibilnosi programa

Ako se program poziva iz mašinca funkcija USR se na manje sa programar sam mora dodaliti odgovarajuću vrednost paru BC Znači

```
LD BC startna adrese
CALL startna adrsea
```

90—120 Početak pretrage

Adresa datoiska što je ujedno i adresa prvog izraza mamorisa se u promanljivoj ADR Ova promanljiva će nadaje ukazivali na adresu izraza koji je na redu za uporađivanja

Treći baji ispod datoteka to jesi IX—3 puni se nulom Ovaj baji je flag koji ukazuja na to da li je sortiranje završeno

140—270 Glavni deo programa Najpre se od 140 do 200 izračunavaju adrese dva susedna izraza koja iraba uporedili Ako sedržaj prvog bajta drugog izraza nija nula upoređivanje može da počnq pa sledi skok na LAB3 (210—230)

Ako je sedržaj nula znači da je program stigao do kraja datoteka Sada se ispituje fleg (240—260) Ukoliko je i njegova vrednosi nula sortiranja je okončano i program završava rad U proiivnom, sledi skok na LAB1 i ponovnu pretragu od početka

290—370 Uporađivanja

Brojač B pmuzima vrednost za dubinu praimge a zatim se dva izraza uporeduju DE ukazuja na adresu prveg a HL na adrseu drugog Ako je (DE) u ma kom stadijumu pretrega manja od (HL) izrazi koji se uporeduju dobro su sortirani pa se dalja ispitivanja prekida i skokom na LAB2 pralazi se na pripremu za upoređivanja drugog para izraza

Ako je (DE) najednako (to jsat veće) izrazi nisu dobro sortirani Zato sledi skok na LAB4 — to je rutina koja će upoređivanje izrazima izmaniti mseta

Ostaja još traća mogućnost, da su (DE) i (HL) jednaki U tom slučaju odluka se ne može doneti DE i HL se inkramaniraju da bi se ispitali sledeći bajtovi To se ponavlja dok brojač na dobija vrednost nula Ako odluka ni tada nija donete znači da su uporađivani izrazi jednaki do dubine ispitivanja — na iraba ih dirati što rezulira skokom na LAB2 i preuzimanjem sledećeg para izraza

Na brzinu sortiranja znatno utiče i to šio program skoro nikad ne uporeduje izraze do poslednjeg bajte — ispiiivanja se napušta čim se naida na prvi par različitih znakova Prilikom sortiranja prezimena Milanković i Milanović na primar ispiivanja se napušia kod četvrtog bajta a kad treba sortirali prezimena Petrović i Milankobić već kod prvog bajta Do kraja se ispituju jedino izrazi koji su potpuno isti (Petrović — Petrović) ali se i to dešava samo u slučaju kada parametri za dubinu pretrage i dužinu izraza imaju jednaka vrednosti

390—550. Izmana mesta

Ovaj deo programa izvršava se samo kad je potrebno da se ispiiivanim izrazima izmana mesta Pošto se ne zna na kojoj je dubini ispiivanja donata odluka o izmani moraju se opet utvrditi početna adresa oba niza a potom i njihova dužina na koju ukazuja (IX--2) Ostatak posla obavlja petlja PT2 (450—530)

Traga o obavljenoj izmani ostavlja se u flagu (IX—3) koji dobija vrednost 1 Zatim se pralazi na novi par izraza skokom na LAB2

Sada postaje razumljivo kako program prepoznaja da je sortiranje kompisana datoteka završeno Program više puta proiazi kroz celu datoteku Na početku svakog proiaza fleg se reseiuje Ukoliko program tokom proiaze obavi makar jednu izmanu fleg se setuja Kad se na kraju proiaza utvrdi da fleg nja setovan to znači da tokom log proieza nije obavljena ni jedna izmana odnosno da više nema šta da se sortira

Žarko Vukosavljević

Uz malu pomoć pameti

U prethodnim „Računarima" započeli smo pisanja programa koji pronalazi rešenje igre „oeem", verzije Lojdovih kvadrata, i finalizovali procedure za unošenje, štampanje i provaru ispravnosti pozicija. Ostalo je da se pozabavimo samim generisanjem rešenja i započnemo pripreme za krupniji programerski zalogaj — igru tic-tac-toe.

Osnove našeg budućeg programa je procedure SOLVE koja sukcesivno isprobava razna varijante sve dok na pronađe korektno rešenja koje će kao što smo videli garantovano biti minimalno U svekom prolazu treba pronaći poziciju čija je vrednosna funkcija minimalna (vrednosna funkcija F (N) je definisana kao G(N)+H(N) gde je G(N) broj poteza koja treba primaniti na početno stanja da bi se od njega dobilo stanja N a H(N) mere udaljenosti stanja N od ciljnog stanja izražana preko broja kvadrata koji u poziciji N nisu postavljani na korektna mesta) Izvršiti njanu analizu izbaciti je iz liste nepoznatih pozicija i nastaviti se redom

Osnovne procedure

Samo pronalaženja pozicije za koju vrednosna funkcija ima minimalnu vrednost obavlja procedura FINDLOW koja je relativno jednostavna — pozicija su dvostruko ulančana u listu koja se analizira od glave prama repu Vrednosna funkcija za svaki čvor mreže se da bi deli rad bio brži upisuja u predviđano polja samog čvora što znači da je ponovni proračun napotrebnan u toku kompletna dalja analiza

Pošto je najperspaktivnija pozicija pronađena treba ja analizirati Ovaj je zadatak poveren proceduri EXPANDLOW koja primanjuja svaki mogući operator na zadatu poziciju I proverava da li je postignuto ciljno stanja Ako jeste tu činjanicu treba registrovati u okviru odgovarajućeg čvora a ako nija novo stanja treba upisati u listu I nastaviti pretraživanje Slika 1 prikazuja atrukturu čvore liste koja je dovoljna za registrovanje podataka koja amo doseda proglselli neophodnima



Stanje čvera

Pokazivač na čvor generisan primenjenim operatorom levo

Pokazivač na čvor generisan primanjenim operatorom gore

Pokazivač na čvor generisan primenjenim operatorom desno

Pokazivač na čvor generisan primenjenim operatorom dola

Pokazivač na sledeći elemenl u listi elemenata

Pokazivač na prethodni element u listi elemenala

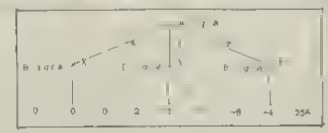
Pokazivač na redlteljski čvor u stablu stanja

Rastojanje od korena stabla

Vrednosna funkcija

slika 1 — Struktura čvora liste za registrovanje podataka.

Nastavljajući analizu „sa vrha na dola pozabavičemo se procedurom TRYOPER koja isprobava operetor Iranaformišući takuća stanja u novo madustanja Procedura je nazvana TRYOPER jer primana zahtevanog operatora nija uvek moguća — ukoliko je na primar prazno polja u donjam lavom uglu na možamo ga pomariti ni na dola ni na lavoi Svaka uspešna primana operatore propraćena ja pozivom procedure DUPLICATE koja proverava da li ja postignuto stanja već u listi tj da li ja dobljano stanja već analizireno III ja u planu za analizu Procedura DUPLICATE bitno ubrzave rad I obazbeduja da rešenja uvek buda pronadano — da nja nama mogio bi se dsalti da računar stalno vrši tranetormaciju izmedu dve ausedne pozicije I tako upedna u mrtvu patljut Ukoliko je najzad utvrđeno da je operator bio moguć I da dobljano stanja nija duplikat nakog od poetojačih poziva se procedura CREATE koja generiše novi čvor u listi I u akladu se alikom 1 ulančava razna ukazalalja U toku ovog kreiranja će prirodno biti izrečunata I vrednoana pozicija postignutog stanja



slika 2 — Glavni program sa osnovnim procedurama

Ostalo je još da pomanamo proceduru DELETE koja uklanja alament liste Uklanjanja je sama dalimično ukazatalji prathodnog čvora (ili HEAD ukoliko se radi o prvom alamantu liata) se na odgovarajući način modifikuju all se podaci na uklanjaju jar su i dalje deo liste mogućih stanja.

Slika 2 prikazuja sva pobrojana procedura par režijskih potprograma (npr PRINT koji štampa pronadano rešenjaj i glavni program koji zapravo predstavlja naznalno proširenu verziju glavnog programa iz prošlih Računare Program ja kao i obično pisan na *Turbo Pascalu 4.0* I testiren na IBM PC AT kompatibilnom računaru Trudili amo se međutim da na primanimo nikak ▶

```
program algli (input, output);

{
        IGRA OSAM

       "Racunari 41"

}

const stranica = 3;

type tablei    = 1 .. stranica;
     pozicija  = 0 .. maxint;
     tabla     = array [tablei, tablei] of char;
     pointer   = ^node;
     node      = record
                     stanje          : tabla;
                     left, up, right,
                     down, nextleaf,
                     prevleaf, parent : pointer;
                     dubina, ocena    : pozicija
                 end;
     operacija = (levo, gore, desno, dole);

var  pocetno, ciljno            : tabla;
     koren, glava, rep, ciljp   : pointer;
     nadjen                     : Boolean;
     dummy                      : string;

   {
   Procedure lepis, ucitaj, izbrisi (iadblank
   swap, moveblank i pariya su objasnjene
   u "Racunarima 40"                              }


procedure heuristika (stanje : pointer; ciljno : tabla;
                      var rez : pozicija);
const blank = ' ';
var row, col : tablei;
begin
   rez := 0;
   for row := 1 to stranica do
      for col := 1 to stranica do
         if  (stanje^.stanje [row, col] <>
               ciljno [row, col]) and
             (ciljno[row, col] <> blank)
         then rez := rez+1
end;

procedure nadjirrea (var novi : pointer;
                     pocetno, ciljno : tabla);
var lmir : pozicija;
begin
   new (novi);
   with novi^ do
   begin
      stanje := pocetno;
      left := nil; up := nil; right := nil; down := nil;
      nextleaf := nil; prevleaf := nil; parent := nil;
      dubina := 0;
      heuristic (novi, ciljno, hsar);
      ocena := dubina + hsar
   end
end;

procedure najbolja (var ekse : pointer;
                    head : pointer);
var  pointer;
begin
   n := head;
   krz := n;
   while n <> nil do
   begin
      if  n^.ocena < ekse^.ocena
      then ekse := n;
      n := n^.nextleaf
   end
end;

procedure trazper (lowest      : pointer;
                   oper        : operacija;
                   rowbl, colbl : tablei;
                   var novo     : tabla;
                   var primenjen : Boolean);
begin
   primenjen := false;
   case oper of
      levo :  if colbl > 1 then
              begin
                 novo := lowest^.stanje;
                 swap (novo [rowbl, colbl],
                       novo [rowbl, colbl-1]);
                 primenjen := true
              end;
      gore :  if rowbl > 1 then
              begin
                 novo := lowest^.stanje;
                 swap (novo [rowbl, colbl],
                       novo [rowbl-1, colbl]);
                 primenjen := true
              end;
      desno : if colbl < stranica then
              begin
                 novo := lowest^.stanje;
                 swap (novo [rowbl, colbl],
                       novo [rowbl, colbl+1]);
                 primenjen := true
              end;
      dole :  if rowbl < stranica then
              begin
                 novo := lowest^.stanje;
                 swap (novo [rowbl, colbl],
                       novo [rowbl+1, colbl]);
                 primenjen := true
              end
   end
end;
function jednakastanja (stanje1, stanje2 : tabla) :
                        Boolean;
var row, col : tablei;
begin
   jednakastanja := true;
   for row := 1 to stranica do
      for col := 1 to stranica do
         if   stanje1[row, col] <>
              stanje2[row, col]
         then jednakastanja := false
end;

procedure duplikata (novo : tabla; root : pointer;
                     var duplikat : Boolean);
begin
   duplikat := false;
   if root <> nil then
      if   jednakastanja (root^.stanje, novo)
      then duplikat := true
      else
      begin
         duplikata (novo, root^.left, duplikat);
         if not duplikat then
         begin
            duplikata (novo, root^.up, duplikat);
            if not duplikat then
            begin
               duplikata (novo, root^.right,
                          duplikat);
               if   not duplikat
               then duplikata (novo, root^.down,
                               duplikat)
            end
         end
      end
end;

procedure kreiraj (novo, ciljno : tabla;
                   lowest       : pointer;
                   oper         : operacija;
                   var novop, rep : pointer);
var vrednost : pozicija;
begin
   new (novop);
   novop^.stanje := novo;
   novop^.left := nil;
   novop^.right := nil;
   novop^.up := nil;
   novop^.down := nil;
   novop^.dubina := lowest^.dubina+1;
   heuristic (novop, ciljno, vrednost);
   novop^.ocena := novop^.dubina + vrednost;
   novop^.parent := lowest;
   case oper of
      levo  : lowest^.left := novop;
      gore  : lowest^.up := novop;
      desno : lowest^.right := novop;
      dole  : lowest^.down := novop
   end;
   rep^.nextleaf := novop;
   novop^.prevleaf := rep;
   novop^.nextleaf := nil;
   rep := novop
end;

procedure expandiranje (lowest, root : pointer;
                        ciljno       : tabla;
                        var rep, ciljp : pointer;
                        var resenje  : Boolean);
var rowbl, colbl         : tablei;
    oper                 : operacija;
    novo                 : tabla;
    primenjen, duplikat  : Boolean;
    novop                : pointer;
begin
   findblank (lowest^.stanje, rowbl, colbl);
   for oper := levo to dole do
   begin
      trazper (lowest, oper, rowbl, colbl,
               novo, primenjen);
      if primenjen then
      begin
         duplikata (novo, root, duplikat);
         if not duplikat then
         begin
            kreiraj (novo, ciljno, lowest,
                     oper, novop, rep);
            upis (novop);
            if jednakastanja (novop^.stanje,
                              ciljno)
            then
            begin
               ciljp := novop;
               resenje := true
            end
         end
      end
   end
end;

procedure delete (var lowest,
                  var glava, rep : pointer);
var temp : pointer;
begin
   if lowest <> glava then
   begin
      temp := lowest^.prevleaf;
      temp^.nextleaf := lowest^.nextleaf;
      if   lowest = rep
      then rep := temp
      else
      begin
         temp := lowest^.nextleaf;
         temp^.prevleaf := lowest^.prevleaf
      end
   end
   else
   begin
      glava := glava^.nextleaf;
      if glava = nil
      then rep := nil
      else
         glava^.prevleaf := nil
   end
end;

procedure resenje (root : pointer;
                   var ciljp, glava, rep : pointer;
                   ciljno : tabla; nadjen : Boolean);
var lowestp : pointer;
begin
   while not nadjen do
   begin
      najbolja (lowestp, glava);
      writeln ('Najmanja vrednost funkcije je ',
               lowestp^.ocena:1);
      writeln ('Sin ovaj da analiziramo: ');
      lepis (lowestp^.stanje);
      writeln ('I tako dobijamo nova stanja: ');
      expandiranje (lowestp, root, ciljno,
                    rep, ciljp, nadjen);
      delete (lowestp, glava, rep)
   end
end;

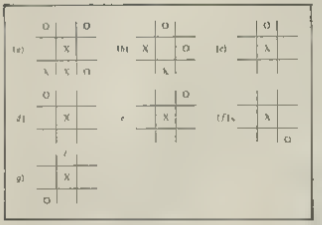
procedure ispisresenja (nodep : pointer);
begin
   if nodep^.parent <> nil then
   begin
      ispisresenja (nodep^.parent);
      writeln ('   ^');
      writeln ('   |');
      writeln ('   |')
   end;
   lepis (nodep^.stanje)
end;

begin { glavni program }
   repeat
      writeln ('Unesite pocetno stanje ');
      unos(pocetno);
      writeln ('Unesite ciljno stanje ');
      unos(ciljno);
      if   not pariya(pocetno, ciljno)
      then writeln ('Resenje ne postoji ')
      else begin
         nadjirrea (koren, pocetno, ciljno);
         glava := koren;
         rep := koren;
         if   jednakastanja (koren^.stanje, ciljno)
         then begin
                 nadjen := true;
                 ciljp := koren
              end
         else
              nadjen := false;
         resenje (koren, ciljp, glava,
                  rep, ciljno, nadjen);
         writeln;
         writeln ('Pritisni CR za resenje');
         readln (dummy);
         ispisresenja (ciljp)
      end;
      writeln
   until false
end.
```

…se posebna specifičnosti *Turbo Pascala* što znači da će program moći da se izvršava i na bilo kom drugom računaru koji je opremljen paskal kompajlerom. Ukoliko ste najzad veliki ljubitelj bejzika, prepravka programa ne bi trebalo da predstavlje poseban problem, obzirom da smo opisali principe rada i funkciju svakog bitnog potprograma.

Pokušajte da potvrdite svoje razumevanje ovoga o čemu smo govorili tako što ćete kompjuterski rešiti odgovarajuću Lojdovu zagonetku iz 1878 zvanu „petnaest” — četrnaest kvadrata na

koja su upisani brojevi 1—14 grupisano ja na tablu 4×4 pri čemu ja jedno polja prazno Zadatek ja lati — transformisati proizvoljno izabranu početnu poziciju u proizvoljno izabranu ciljnu (rešenja jasno na mora uvek postojati) primenjujući minimalni mogući broj poteza Primetićete da su promene u programu minimalna



Slika 4 — Stablo zamišljene igre

ali da je vreme razmišljanja bitno duža što nija teško objasniti značajnim povećanjem broja mogućih pozicija

Tic-Tac-Toe

Sledaća tema ova mala serija tekstova ja igra *tic-tac-toe* čija pravila ne treba posebno objašnjavati — dva igrača naizmenično ateviljaju X i O ne tablu 3×3 pri čemu se svaki od njih trudi da se na nekoj horizontali dijagonali ili vertikali tebla nadu tri njegove znaka Ukoliko ni jedan od igrača ne pogreši partija se završava remijem dok su u praktičnim uslovima šanse prvog igrača nešto veće zbog semki koje može da postavi (uokolo na primer prvi znak buda smeštan u neko od ugaonih polja drugi igrač gubi ukoliko odmah na popuni centralno polja ako prvi igrač popuni centralno polja drugi gubi ako sa ne odluči se ugaono polja itd)

U poređanju sa drugim logičkim igrama koje sa ne zasnivaju na alamantu sreće (šah go dama) igra *tic-tac-toe* ja relativno jednostavna se kompjuterskau analizu i obradu — sasvim je najasno zašto ja kompjuteru u Ratnim igrama bilo potrebno onoliko vrama da shvati u čemu je stvari Pojavljuja se ipak nakoliko novih momenata od kojih ja najznačajnija vrednosna funkcija — treba ja definizati tako da realno odalika vrednost pozicija a da ipak na buda prekomplikovana se izračunavanja Prilikom analiza pozicija laprobavačemo sva moguća poteza prvog igrača a zatim i sva moguće odgovora drugog Vrednosna funkcija *F(N)* mogla bi da se definiše kao +100 ako prvi igrač ppobeduja −100 ako prvi igrač gubi i u ostalim slučajevima, (*broj linija koja su otvorene za prvog igrača) minus (broj linija koje au otvorene za drugog igrača).* Za prve dve pozicija sa slika 3 ja na primar *F(N)* 100 odnosno nula dok će vam preostali primari poslužiti kao dobra vežba. Poslednja četiri pozicija se slika 3 su almatrična što bi značilo da vredi ispitivati samo jednu od njih — na taj način ćemo bitno skratiti vreme računanja jar će u početku partija biti mnogo almatričnih posicija a na kraju malo mogućih poteza

Pisanja programa *tic-tac-toe* sehteva razumevanja maloda *običnog* (ovde se uglavnom bavimo hauristićkim) pretraživanj ma koja se naziva *minimaks* iako arno o minimaks postupku govorili u okviru naše „Radionice logičkih igara (Računari 9, 11 13) dapustićemo sebi da ponovimo osnovne atvari varuyući da arno za ove tri godina atekli dosta novih čitalaca

Minimaks postupak

Posmatrajmo atablo neka semišljana igre prikazano ne slici 4 Vidimo da se igra približila kraju i da atvari atoja vrlo loša za igrača A u jednoj od grana ga doduše očekuje apaktakularni dobitek od +256 poana ali au ostala alternative gubitnička Pokušajmo da izaberemo jedan od tri polaza koje A može da odigra sledeći isključivo nehaurističku logiku koju ja bliska kompjuterskoj

Sva tri poteza koje bi B odigrao u poziciji X vode do remija što znači da bi pozicija X bilo umesno pridružiti vrednost 0

Sva tri poteza koja bi B odigrao u poziciji Y dovoda do gubitaka za A. pitanja ja samo se kolikom raslikom B dobija Jasno ja da će pozicija Y biti pridružan naki negativni broj ali ćemo as odlučiti koji ja to broj tek kada razmotrimo pozicija Z

U poziciji Z igrač B ima na raspolaganju tri potasa od kojih jedan vodi u spaktakularnu pobedu igrača A Razmatrajući pozicija Z igrač A traba da razmotri kolika ja varovatnoća te pobede Ako A amatra da će B igrati potpuno slučajno (ne primar bacajući kockicu), bilo bi logično uavojiti (−9+(−4)+256)/3=81 za vrednost pozicija Z što znači da bi A avakako izabrao potez Z a na potez X Madutim rakli amo da razmatramo igra u kojima boginja Fortuna nema mnogo uticaja na tok stvari B će razmotriti pozicija i isabrali potez koji mu bude izgledao najparspektivnija Ne mora se jasno dogoditi da lo buda i objaktivno najbolji potez ali ja najalgurnija pretpostaviti da će B odigrati onaj potez ojl bi A odigrao da ja na njegovom mastu To znači da ćemo pozicij Z dodaliti vrednost −9 tj minimum vrednosti −9 −4 i +256 Na alićan način pozicij Y dodaljujamo minimum vrednosti −2 −1 i −7 tj broj −7

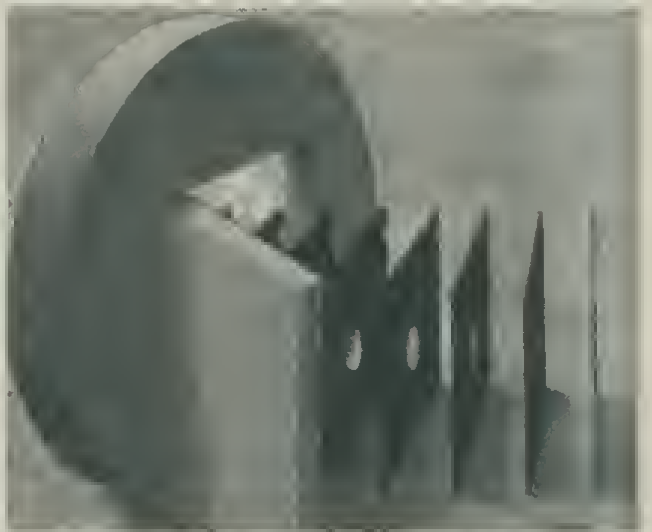
Pošto smo razmotrili pozicija X Y i Z isabraćemo onu koja nam nudi **najveći** mogući akor pri optimalnoj igri protivnika Obzirom da ja makaimum brojava 0 −7 i −9 upravo 0 odlučićemo se as potez X koji nas vodi do nerešene partija Da bi ava bilo sasvim jasno ponovićemo aksioma koja računar treba da sledi da bi izabrao najbolji mogući potez sledeći minimaks postupak

1 Vrednost pozicija u kojoj ja A na potezu je **maksimum** vrednosti pozicija koje se dobijaju odigravanjem svakog od mogućih poteza

2 Vrednost pozicija u kojoj ja B na potezu ja **minimum** vrednost pozicija koja se dobijaju odigravanjem avakog od mogućih poteza

Kada se minimaks postupak izloži na ovaj način reklo bi sa da nlčega jednostavnijaga na avetu nema akoro da bista mogli da sednete za kompjuter i da počnala da pišete novi Coloseus šah! Pra nego što sastavimo program koji ilustruja minimaks postupak dužni amo međutim da izložimo i njegove nedostatke koji ga čina nauppotrebljivim kod tola složenijih igara

Činjanica da sa avakoj poziciji primanom minimaka matoda pridružuja konačen broj koji predstavlja njanu vrednost znači da ja vrednost svake pozicija unapred odlučena ukotiko ja pozicija dobila vrednost +5 A će sigurno pobediti aa bar pet posna prednoati što znači da partiju nama mnogo amtala Igrati Obzirom da ja šah jedna od igara za koju ja **moguće** sastaviti minimaka atablo javite Karpovu i Kasparovu da as bez potrabe muče igrajući mačave! Navolja je međutim što ja za primanu minimaks postupka potrebno posedovati stablo **čitavog** toka igre da bi se odredila vrednost neka pozicija Možata da zamislite da ja stablo šahovska igre **prilična** (ali kolika?) velika i da ga ne bi bilo lako nacrtati



Možda bi ipak vredelo probati? Pra nego što se odlučite se taj pokušaj treba da znata da šahovako atablo ima oko 10^120 pozicija. Imata i predatavu o ovom džinu od broja? U jednom veku ima samo 10^19 **nanosekundi** a sa avaku od tih mikrosakundi čak i najbrži kompjuter današnjice može da izvrši oko 1000 oparacija kao što je sabiranja i oduzimanja Kada bi se samo jednim aabiranjem mogla generisati nova šahovaka pozicija za ispitivanja 10^120 pozicija bi trebalo oko 10^100 **vakava** A vaalona ja stara mnogo manja **sakundi** Vrlo je dakla naverovatno da ćemo u zamislivoj budućnosti imat prilika da procenimo početnu poziciju šahovske igre kada to budamo mogli očekujte šahovski prablam na koma je nacrtana početna pozicija i ispod koga piše bali vuče i dobija u najviše 606 poteza

Rešenja je naravno u haurističkom pretraživanju koje će obezbediti da se pretražuje samo mali ali i najparspektivniji segmant stabla razuzimajući naravno da najbolji mogući potez hudie propušten atepen rizika direktno zavisi od kvaliteta izabrana vrednosna funkcija U sledećim Računarima procenićemo kvalitet *tic-tac-toe* vrednosna funkcija koju smo izabrali i započeće plaanja programa trudeći se pri toma da stablo igra do maksimuma potkrašemo i

Dejan Ristanović

Klub programera

Bejzik/„spektrum"

Ukrštanje na ekranu

Ako se zanimate enigmatikom evo prilike da pravite ukrštene reči bez dosadnog iscrtavanja kvadratiće bez hartije olovke i obavezne gumice za brisanje. Prekucajte bejzik sa priloženog listinga i startujte ga sa RUN. Na ekranu ćete ugledati mrežu za ukrštenicu i svetli kursor u gornjem levom uglu.

Kursor se vodi dirkama sa strelicama (5 = levo 6 = dole 7 = gore 8 = desno). Pritiskom na nulu ucrtava se crno polje na mestu kursora a pritiskom na 9 izlazi se iz programa.

```
 10 REM ----------------------
                  UKRSTENICA
 20 REM ----------------------
 30 POKE 23658 8  PRINT
 40 FOR n=1 TO 15: PRINT CHR$ (n
+64)      NEXT n
 50 FOR n=2 TO 20 STEP 2  PRINT
AT n 0 CHR$ (n/2+64)  NEXT n
 60 LET y=164  LET x=12
 70 FOR n=1 TO 11
 80 PLOT 12 y  DRAW 240 0
 90 LET y=y-16
100 NEXT n
110 FOR n=1 TO 16
120 PLOT x 144: DRAW 0 -160
130 LET x=x+16
140 NEXT n
150 REM ------------------
160 LET x=2  LET y=2
170 BEEP  02 19  PRINT AT y x: 0
BRIGHT 1; OVER 1;
180 IF INKEY$=   THEN  GO TO 180
190 PRINT AT y x: OVER 1;
200 IF INKEY$= 8  THEN  LET x=x+
2  IF x>30 THEN  LET x=2
210 IF INKEY$= 7  THEN  LET y=y
2  IF y<2 THEN  LET y=20
220 IF INKEY$= 6  THEN  LET y=y+
2  IF y>20 THEN  LET y=2
230 IF INKEY$= 5  THEN  LET x=x-
-  IF x<2 THEN  LET x=30
240 IF INKEY$=    OR (INKEY$>= A
 AND INKEY$<= Z ) THEN  PRINT AT
y x INKEY$
250 IF INKEY$= 0  THEN  PRINT  I
NVERSE 1;AT y x;"
260 IF INKEY$= 9  THEN  GO TO 20
0
270 GO TO 170
```

Ostalo pogađate da biste upisali neko slovo treba samo da pritisnete odgovarajuću dirku. SPACE briše upisani znak ali ako hoćete našto da prepravite brisanje nije obavezno dovoljno je da preko starog znaka ukucate novi.

Program je to sta i sami zaključili napisan na spektrumovom bejziku. Vlasnicima drugih računara neće biti teško da ovih dvadesetak programskih redova prilagode svojim mašinama.

Prionite dakle na posao uz enigmatsku devizu Ukrštajmo reči a ne mačeve.

Žarko Vukosavljavić

Bejzik/„spektrum"

Nizovi slučajnih brojeva (1)

Verovatno ste se već susretali sa potrebom generisanja niza međusobno različitih slučajnih brojeva. Takođe vrlo verovatno u tome i ne vidite neki veliki problem zar ne? Generisaćete slučajan broj koristeći standardnu RND funkciju uporediti ga sa prethodno generisanim i bebaciti ako je jednak sa nekim od njih.

Zvuči krajnje logično ali da vidimo šta se događa kada se recimo vaša simulacija TOMBOLE polako približava kraju. Vreme proveravanja potencijalnog kandidata za novi slučajni broj raste što možda i nije tako strašno ali je poprilično poražavajuće što dozvoljavate da vaš računar smišljajući slučajne brojeve lutara po celom opsegu brojeva iako mu se krug mogućnosti sužava. Da ne spominjemo ekstremnu situaciju izvlačenja poslednjeg broja kada će se poprilično namučiti a vi poprilično nečekati iako je preostala jedna jedina vrlo očigledna mogućnost. Očigledno je da za ovakvu primenu najočiglednija rešenja nije previše racionalno. Evo i drugog načina.

Definišemo pomoćni niz a() koji ima onoliko elemenata koliki je opseg biranja i promenljivu **opseg** koja sadrži upravo tu vrednost (npr za simulaciju TOMBOLE sa 90 brojeva ovaj niz će sadržati 90 elemenata a promenljive **opseg** sadržavaće vrednost 90). Na samom početku generisanja slučajnih brojeva svaki element niza a() biće jednak svom indeksu tj a(1) = 1 a(2) = 2, a (25) = 25. U ovoj varijanti rezultat RND funkcije tj INT(RND·opseg + 1) ne uzimamo kao mogući slučajni broj koga treba proveravati već kao indeks elementa niza a() čiju vrednost uzimamo kao traženi izabrani slučajni broj. Jasno u prvom krugu izvlačenja rezultat i prve i druge metode je isti. Razlika nastaje kada na mestu ovog izabranog elementa niza upišemo vrednost elementa **a(opseg)** tj momentalno poslednjeg a promenljivu opseg umanjimo za 1.

Uzmimo primer porspomenute TOMBOLE OD 90 brojeva. Na početku niza a() izgleda ovako.

Neka je recimo rezultat izraza INT(RND·opseg + 1) broj 8. Prvi izabrani slučajni broj biće a(8) tj baš sam broj 8 član a(8) će poprimiti vrednost elementa a(opseg) tj u ovom krugu izvlačenja to je a(90) (u sledećem će biti a(89) pa a(88) itd) Niz a() će dakle izgledati ovako

a(1)	a(2)	a(8)	a(88)	a(89)	a(90)
1	2	8	88	89	90
		OPSEG = 90			
a(1)	a(2)	a(8)	a(88)	a(89)	a(90)
1	2	90	88	89	90
		opseg = 89			
a(1)	a(2)	a(8)	a(88)	a(89)	a(90)
1	2	90	89	89	90
		opseg = 88			

```
10 REM - primer  TOMBOLA sa 90 brojeva·
20 DIM a(90)   REM·pomoćni niz·
30 DIM b(90)   REM·niz u koji redom smeštamo generisane sluč. br·
40 LET opseg = 90
50 FOR I=1 to opseg   LET a(I)=I NEXT I   REM·inicijalizacija·
60 FOR J=1 TO 90   REM·biće izvučeni svi brojevi·
70 LET z= INT(RND opseg + 1)
80 LET b(j)=a(z)   LET a(z)=a(opseg)   LET opseg=opseg 1
90 NEXT J
```

Pošto smo smanjili vrednost promenljive **opseg** koja figurira u izrazu sa RND sada se mogu birati samo elementi niza od a(1) do a(89) među kojima nijedan nema vrednost već izabranog slučajnog broja tj 8.

Stvar se bitnije ne menja ni u situaciji koja na prvi pogled može izgledati sumnjiva — kada rezultat RND izraza pokaže baš na poslednji element u datom opsegu tj konkretno u ovom našem slučaju u drugom krugu izvlačenja član a(89). On će se iskopirati na isto mesto a ponovno smanjenje promenljive **opseg** jednostavno će ga isključiti iz ponovnog izbora.

Na ovaj način svi traženi slučajni brojevi generisaće se istom brzinom sigurno će biti međusobno različiti i još našto ova metoda je ne samo neuporedivo brža već zbog stalnog pretumbavanja članova izgleda i za nijansu RANDOMNIJA od prvospomenute.

Međutim ovo ne znači da prvu ideju treba zaboraviti jer neophodna inicijalizacija niza a() može da „pojede" vremensku uštedu ove metode ako se radi o trkama na kratka staze tj kada se generišu manji nizovi slučajnih brojeva odnosno nizovi sa malim brojem članova.

Damir Omrčen — OCKY

Bejzik/„tim 011"

Nizovi slučajnih brojeva (2)

Mućkanje kuglica za LOTO i TOMBOLU i mešanje karata su lepi primeri upotrebe niza različitih slučajnih brojeva pa ćemo se još malo zadržati na generisanju takvog niza. Uporedićemo Damirov program i jedan program koji sam ja koristio u slične svrhe.

Drugi program pored toga što je kraći uzima i manje memorije jer dimenzioniše samo jedan niz. Pokazao se bolje i na brzinskim testovima (9 5 sekundi prema 14 5 na redakcijskom timu). Postavlja se jedino pitanje da li je dovoljno da se izvrši onoliko izmena članova niza koliko niz ima elemenata. To u

```
  5 REM Damir
 10 OPSEG = 1000
 20 DIM A(OPSEG)
 30 DIM B(OPSEG)
 40
 50 FOR I=O TO OPSEG
 60 A(I)=I
 70 NEXT I
 80
 90 P=OPSEG
100 FOR J=O TO P
110 Z=INT(RND·(P+1))
120 B(J)=A(Z)
130 A(Z)=A(P)
140 P=P-1
150 NEXT J
160
170 FOR I=O TO OPSEG
180 PRINT B(I)
190 NEXT I
200
```

```
  5 REM Voja 2
 10 OPSEG=1000
 20 DIM A(OPSEG)
 30
 40 FOR I=O TO OPSEG
 50 A(I)=I
 60 NEXT I
 70
 80 FOR I=O TO OPSEG
 90 SWAP A (RND'OPSEG)
    A(RND 'OPSEG)
100 NEXT I
110
120 FOR I=O TO OPSEG
130 PRINT A(I)
140 NEXT I
150
```

velikoj meri zavisi od sistemskog generatora slučajnih brojeva.

Ako želite da se osigurate i povećate broj izmena možete da uvedete i faktor mešanja (K) i prepravite drugi program ubacujući linije

```
25 K=2
80 FOR I=1 TO OPSEG K
```

U tom slučaju izvršiće se dvostruko više izmena nego što ima članova niza. Promenljivoj K možete naravno da dodelite bilo koju vrednost (veću od nule) i tako određujete koliko dugo ćete mešati karte.

Voja Gašić

Bejzik/„komodor"/„spektrum"

Analizator Bulovih iskaza

Problem je poznat treba napisati program koji pomaže rešavanju složenih logičkih iskaza crtanjem tabele iz koje je moguće za razne vrednosti promenljivih očitati odgovarajuću logičku vrednost iskaza. Korisnik bi trebalo ne ulazu da zada iskaz a program da uradi sve ostalo tj štampa odgovarajuću

```
10 REM ************************
20 REM *                      *
30 REM * Analizater Sulovih   *
40 REM *       leksiza        *
50 REM * Cerovski Viktor      *
60 REM *   (c) 7  V) 1988     *
70 REM *                      *
80 REM ************************
90 REM
100 DIM IR(100): DIM m$(1): DIM v(15): DIM nm$(5)
110 INPUT "Izraz) ";IR$: LET s=0: PRINT "Izraz) ";IR$
120 FOR i=1 TO LEN IR$
130 LET m$=IR$(i)
140 IF (m$< "a" OR m$>"z") AND m$<>"+" THEN GO TO 200
150 LET nm$=nm$+ m$: IF m$="+" THEN GO TO 170
160 LET s=s+1: LET nm$=nm$(1 TO s): PRINT nm$
170 FOR j=1 TO LEN IR$
180 IF IR$(j)=m$ THEN LET IR(s)=IR(s)+nm+IR(j+1 TO )
190 NEXT j
200 NEXT i: PRINT "Vrednost"
210 FOR i=0 TO s: LET v(i)=0: NEXT i
220 FOR i=1 TO 255
230 FOR j=1 to s: PRINT v(j);" "; : NEXT j
240 PRINT "Val IR$
250 LET j=1
260 LET v(j)=NOT v(j): IF v(j)=0 THEN LET j=j+1: GO TO 260
270 NEXT i
```

tabelu. Varovalno se sećala programa iz Računara 36.

Na ulazu našeg programa je neki logički izraz. Kako izračunati njegovu vrednost? Pa to već zna svako dete: treba izraz pretvoriti u postfiksni ili prefiksni oblik i zatim ga u takvom obliku izračunati. Kako do ovih oblika izraza doći pisano je u Računarima više puta (naravno ako znamo neki bolji (brži/kraći/elegantniji) način da rešimo problem iskorišćenjem ga dok originalnost po svaku cenu obično na daje dobra rezultate). Tako napisan program u bejziku na zauzima previše memorije (ne traba mu čak ni 65520 bajtova za stek) i lako se može napisati i na Galaksiji.

Problam možemo rešiti i lako što će kompajlari umesto da se mi semi trudimo da napišemo program koji računa vrednost izraza te uradiih sem imajući u vidu da kompajlari i interpreteri obično imaju ugrađene potprograme koji upravo to rade. Taka na primer spektrumov bejzik ima naredbu VAL koja može da izračuna vrednost šitavog izraza koji joj dostavimo kao argumenti u nekom stringu a komodor ima -Spakirum simulatior. Vrlo zgodno zar ne? Sad naš Analizator Sulovih izraza na Spektrum simulatoru (ili semom spektrumu, ako ga imate) može da izgleda ovako:

1) Učitamo string koji predstavlja naš logički izraz. Izraz se zadaje lako što se za promenljive koriste proizvoljne male slova a pet osnovnih logičkih operacija predstavljamo simbolima + · NOT = <= (= predstavlja ekvivalenciju a <= implikaciju). Moguće je naravno koristiti i praktično proizvoljan broj segrada. Pri plasnju izraza traba voditi računa o tome da implikacija i ekvivalencija (za program) imaju prioritet veći od disjunkcije a manji od konjukcije (razlog za ovu nelogičnost objasnićemo kasnija) što nija poseban problam jer na kod složenijih izraza svejedno koriste segrada.

2) Izraz modifikujemo tako da možamo da kontrolišemo vrednosti promanljivih. To se radi tako što eko se npr promanljive q e i p (ako se pojavljuju u izrazu) zamanjuju se v(1) v(2) i v(3) respaktivno. Sada se manjanja vrednosti prve promanljive iz izraza svodi na promenu vrednosti v(1). Traba imati u vidu da je vrednost za lacno u spaktrumovom bejziku 1 (a ne −1 kao u većini drugih bejzika — baš zato je implikacija predstavljana znakom <=). Da li sad možama se VAL da izračunamo za konkratne vrednosti promanljivih (logičku) vrednost izraza? Na još jer problema pravi disjunkcija koja je zamanjana sabiranjem a 1+1=2 što je izvor problema pa se operacija + zamanjuje (logičkom) operacijom OR. Tima je nastao jedan a pojavio se drugi (ranaćom manji) problam jer OR u odnosu na (našu) implikaciju i ekvivalanciju ima manji prioritet i ovo bi se mogio lako rešiti — da se tim ima poirabe.

3) Sad kad je konačno izvršena modifikacija polaznog izraza možemo da lapisujemo tabelu lako što ćemo radom odradivati sve moguće vrednosti za nazavano promanljive iz izraza i za svaku varijaciju vrednosti izračunati (funkcijom VAL) i odgovarajuću vrednost izraza. Da je naš izraz slučajno ainiskano nalaspravan bila bi prijavljena ainiskana greška pri izvršavanju VAL naredbe.

Taka smo dobili analizator Sulovih izkaza u 19 linija bejzik programa (par stotina bajtova memarija).

Ako ste aračni visanik nakog PC-a a treba vam baš ovakav program (a treba vam jer nija igra već koriatan program a PC je korisne mašinaj predlažam vam da uradite jednu od sledećih stvari:

1) Sledeći uputstva iz ovog iskaza pratljaste starija brojeva Računara. I nadate ponešto o RPN-u i slično stvarima a zatim i napišite program (obavazno na paskalu). Koriano jer možete i nešto nova da naučite.

2) Napišite Spektrum simulator za PC — nikad se ne zna kakva će sva programa pisati za PC zato je bolja obezbediti se na vreme.

3) Prevedite program na „True Basic" na primar Koriano jer je to najbrži način da dodate do Analizatora na PC-u.

4) Prakucajte sličan program iz Računara 38. Koriano jer ćete naučiti kako se kompajlaru zadaje veličina steka (ako namate upuitstvo za „Turbo Pascal") a ti modarni editari sve nakakvi meniji i apolja čovek se prosto izgubi. A nakade je bilo dovoljno aikucati nešto poput STACK XX.

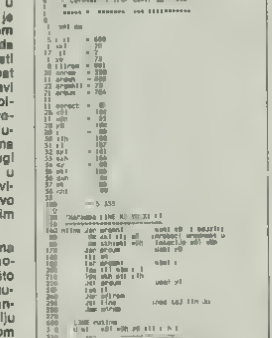
Viktor Cerovski

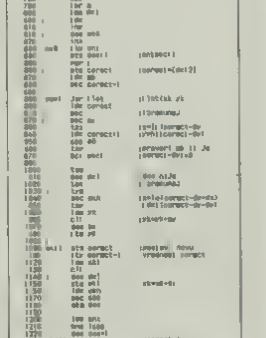
6810/„komodor 64"

Crtanje linija

Jedan od najvažnijih grafičkih potprograma je svakako potprogram za crtanje linija (duži) prvenstvano zbog primena u obiasti vektorske grafike. Za dobijanja potprograma poslužićemo se algoritmom Milaia Čekovića iz Računara 31, na osnovu koga je napisan priloženi program u asamblaru.

Novedobijana LINE naredba je brža pri crtanju dugih linija od odgovarajuće naredbe iz Simon a basica oko 2.5 puta. Dalje ubrzavanja ovog kao i ranije objavljanih programa moguće je postići ako imamo u vidu zgodnu osobinu korišćenih algoritama da se pri uzastopnim linearnim lacrtavaju susedne tačke pa bi mogli da na osnovu adrese prethodna tačke izračunamo adresu sledeće. Svakako najvažnije prednost ovakvog načina kodiranja je povećana brzina izvršavanja potprograma ali evakve tehnika ima i više mana. problami obično nastaju kad dao objekta kojeg crtamo (npr. elipse) izlazi izvan ekrana — detektovanja ovakvih slučajeva obično se na iskodira (jer to zahteva dodatna instrukcija — dodatno vrama za izvršavanja) rutina obično radi samo u jednom grafičkom modu a kodiranja zahteva više vremena jer je za dobijanje





Da bismo dobili i nove bejzik naredbe potrebni su ranija objavljeni programi u Računarima u kojime treba promeniti linija

```
4200 naredbi=5
4210 esc  circle ellipse line
4220 wor A1,A2,$1333
```

Podrazumeva se da je unasen i program iz Računara sa naredbama za crtanja krugova i elipsi (meda nija neophodna).

efikasne rutina potrabne prečizne programiranja u asamblaru a takođa i dobro poznavanja organizacija video-memorije. Potpisniku ovog iskaza se čini da je bolja — ako smo već odlučili da napišemo program — potrošiti vrama na amišljanja što efikasnija algoritma nego na što efikasnija kodiranje.

Viktor Cerovski ▶

▶ 6810/„komodor 64"

Konverzija dec-hex-bin

Program vrši pretvorbu dacimalnog broja u hakadakadski i binarni oblik. Na ulazu na adresama 251/252 se menju naisalti niži i viši bajt broja koji se konvertira

Broj se ispisuje u obliku
dec=$hex=%binH—binL

Ukoliko se na adresi 252 nalazi vrijednost 0, viši bajt binarnog oblika se ne ispisuje

```
Program  DEC-HEX-BIN KONVER-
ZIJA
Sintaksa  SYS 10000
*
Ulaz  Na 251/252 se nalaze 10- i hi-
bajtavi broja

          *=10000

low     =251
high    =252
pom     =253

        ldy low
        lda high
        jar 48589
        lda #  =
        jar $ffd2
        jar hex
        lda #  =
        jar $ffd2
        jmp bin

hex     lda #  $
        jar $ffd2
        lda high
        jar t01
        lda low
101     pha
        lar a
        lar a
        lar a
        lar a
        jar 102
        pla
        and #16
102     cmp #10
        bcc 103
        adc #8
103     adc #  0
        jmp $ffd2

bin     lda #  %
        jar $ffd2
        lda high
        beq c1
        jar 105
        lda #  -
        jar $ffd2
c1      lda low
105     sta pom
        lda #0
        ldy #9
104     jar $ffd2
        asl pom  bit u carry
        lda #  0
        adc #0  a=chr$(48)+carry
        dey
        bne 104
        rts
```

U haka modu bajt se množi sa 16 (niz ASL A naredbi, 214=16), zatim se provjerava da li je veći od ASC(9). Ako jeste, dodaje se 7 (6+Carry=AS-C(A)+7=ASC (.A) i ispisuje se

U bin rutini vrši se logičko pomjeranje ulijevo. Bit koji je bio sedmi u pomoćnoj lokaciji se pronosi u keri flag a ovaj se dodaje na ASC(„0") i ispisuje

Tomo Sombolac

6810/„komodor 64"

Vodeće nule

Ponekad je u programima koji rade s brojevima ili u igricama zgodno imati mogućnost poravnavanja udesno i lepih vodećih nula. Ova rutina ispisuje brojeve na taj način. Prije pozivanja potrebno je na adrese 251/252 upisati niži i viši bajt broja koji želimo ispisati

U programu je upotrijebljen jedan trik na adresi 48589 (Basic-ROM) sa nalazi rutina za ispis 16-bitnih brojeva. Na izlazu iz te rutina Y registar sadrži broj znamenki ispisanog broja. Da bi se program skratio, prvo se otvara lažni izlazni uređaj (ovdje kazetofon) i u bafer se upiše broj. Zatim se ispisuju nule do peteroznamenkastog broja (6 umanjeno sa vrijednost Y registra). Poslije se otvara standardni izlazni uređaj (ovdje ekran) i broj se ispisuje

Važno je napomenuti da se lokacija 166 mora brisati tj postavljati na broj manji od 192 jer je to brojač znakova u baferu kazetofona pa bi pri prekoračenju navedenog broja operativni sistem ispisao PRESS REC & PLAY ON TAPE

```
*Program  Vodeće nule
*Sintaksa  SYS 49152
*
*Ulaz     na 251/252 je
*broj

          *=49152

low     =251
high    =252
pom     =253
znak    = 0
dev     =3

lspis   lda #1      ,lažni izlazni
                    uređaj
        sta 154
        sta 166     briše bafer
                    kasetofona
        jar lspbr   broj znamenki
                    broja
                    se smješta u Y
        lda #dev    standardni
                    I/O uređaj
        sta 154
        lda #znak
11      cpy #5      5-znamenkasti
                    broj?
        bcs lspbr   da lspiai ga.
        jer $ffd2   lspisuje nulu
        iny
        bne 11
lspbr   ldx low     niži bajt
        lda high    viši bajt
        jmp 48589   lspisuje broj
```

Tomo Sombolac

Z80/„amstrad CPC 464"

Osmobitni interfejs (1)

Kupovinom štampača DMP—4000 poželeo sam da praka Centronica interfejsa pošaljam i osmi bit ali to, na žalost, na amstradu nije tako lako. Rešenja sam našao u članku objavljenom u Računarima 15 pod nazivom „Paralelni interfejs za štampač" autora Svetislava Zahara na strani 54. Pun slana odlučio sam se sa hardvarsku preprevku spajanja 12-tog pina IC—8255 sa devetim pinom konaktora za štampač. Uz malo muke i puno neizvesnosti oko dobrog lemljana prešao sam posla sklapanja računara na deo posla oko učitavanja odgovarajućeg softvera sa podrškom osmog bita

Razočarenje Pokušaj bilo kakvog štampanja dovodi do kraha

Zaključak ← Članak je varovatno pisan sa CPC-464 kod koga postoji razlika u rutinama smeštanim u ROM-u

Posledica ← Rutina MCBUSY PRINTER se nalazi na adresi &0848. Programaku liniju 60 sa al 0 treba prepraviti da glasi

```
60 BUSYPR  EQU #0849
```

Oduševljenje ← Nema kraha. Prvi pul ugledah kurziv crno na belo!

Razočaranje ← Opet krah. Kontrolni kod sedam daje bip ali ne i povratak u bajzlk

Zaključak ← Nešto na vaija

Posledica Programaku liniju 260 sa iste slike treba prepraviti da glasi

```
260 DALJE  LD B #EF
```

Oduševljenje ← Svira svira i u bajzlk sedana

Meditacija ← Kad na može brže onda može krada — prepraviti liniju

```
379 xor 9
```

Razočarenje ← Pokušaj hex-dumpa slanjem kodova od 0 do &FF nija uspeo

'a) Umesto da štampa A0 A1 A2 A3. A6 .AB AC AD AE AF

b) Štampeo je na trepnuvši 5E 5C 7B 23 40 7C 7D 7E 60 5B

Zaključak ← Pregledati print rutina ROM-a *Posledica* ← Unutar rutina MC PRINT CHAR postoji petlja koja pre poziva rutina MC WAIT PRINTER proverava kod znaka sa štampanje. Ako je kod identičan nakom od kodova je reda pod (a) biće zamanjan odgovarajućim kodom iz reda (b) Pra startovanja programa neophodan POKE &b804 9

Oduševljenje Bilo bi potpuno da štampa kompletan tax damp bez automatskog CR LF na svakih 132 adštampanih znakova

Razočarenje ← Potpuno!

Zaključak ← Čemu onda printer A3 formata kad na može više od 132 znaka? *Posledica* ← Prepraviti programske linije

```
390    INC A
400    LD (#AC08) A
410    POP AF
420    POP BC
430    SCF
440    RET
```

Oduševljenje. ← Sada, kada eve radi onako kako je trebalo da radi još kada je prvi put program preslan iz Računara tačno je šest ujutru pa sam zaista i suviše umoran da bih sa radovao. Užitak rada sa svih osam bitova osteviću sa kasno popodne. A saradnicima Računara savet triput marite i kad izmarite onda šaljite prilog časopisu

Aleksander Kosanović

Z80/„amstrad CPC 6126"

Osmobitni interfejs (2)

Vlasnik ste računara CPC 6128 i hardverski ste lemanili interfejs za slanja osmog bita (prama „Paralelni interfejs za štampač" — Svetislav Zahar „Računari 15/86). Unali ste mašinski program ali na radi ili ste nabavili program AMSWGEN koji koristi Zaharov program i to takođe na radi. U mašinskom programu se nalazi greška: labela DALJE LD 8C, #EF a ispravno je DALJE LD 8, #EF. Ovaj tapsus je ispravljen u AMSWGEN-u. Mašinski program radi na CPC 464 ali na i na CPC 6128. Jer sa razlikuju adresa slatamske rutina BUSY PR. Kod CPC 464 je na #81B a kod CPC 6128 je na #858. Ako ispravite ovu razliku AMSWGEN radi koraktno. Međutim ako ovaj mašinski program koristite sa osmobitnu grafiku, neophodna su dva POKE-a. Ukucajte program

```
10 PRINT #8,CHR$ (27), " ,
CHR$(9) CHR$(44) CHR$(1),
30 FOR I=1 TO 300
40 PRINT
#8,CHR$(&(&X10101111)
50 NEXT I
60 PRINT #8
70 PRINT #8, CHR$(27)
```

program za 8-bitni interface

```
        ORG     #A000
        PUSH    HL
        LD      HL,#8804        isključuje konverziju ASCII kodova
        LD      (HL) 0          od #A0 do #AF
        LD      HL,#AC09        sprečava slanje
        LD      (HL) #FF        napoželjnog CR/LF
        LD      HL,waitpr
        LD      (#BDF2) HL      nova adresa skoka u JUMP BLOCK-u
        POP     HL
        RET
busypr
        EQU     #858
waitpr
        LD      BC,#32
ul2     CALL    busypr
        JR      NC sendpr
        DJNZ    ul2
        DEC     C
        JR      NZ,ul2
        DR      A
        RET
sendpr
        PUSH    BC
        LD      B #F6           kanal C IC 8255
        BIT     7 A             karakter veći od 127
        JR      Z dalje         ako nije preskoči
        PUSH    AF
        LD      A,#20           5 bit setovan
        OUT     (C),A           slanje podatka na kanal C
        POP     AF
dalje   LD      B #EF
        AND     #7F
        OUT     (C) A           podatak bez STROBE na izlaz
        OR      #80
        DI
        OUT     (C) A           STROBE on
        AND     #7F
        EI
        OUT     (C),A           STROBE off
        PUSH    AF
        LD      B #F6           kanal C IC 8255
        LD      A,0
        OUT     (C) A           brisanje 5. bita kanala C
        POP     AF
        POP     BC
        SCF
        RET
```

Ovim ste zadali da se odštampa 300 vertikala 10101111 (&AF) u grafičkom modu 0 (na početku reda je slovo „H', da bi ste lakše utvrdili početak vertikale) Slika 1 je rezultat rada ovog programa- a tu nešto nije u redu Odštampana su vertikale 01011011 (&58) na dva mesta se nalaze bajtovi 00001101 i 00001010 (#0D &0A — CR/LF) i na kraju linije su četiri simbola C Računar vrši konverziju nekih ASCII kodova od &A0 do &AF prema printer translation table a posla 132 poslata koda šalje i jednu CR/LF sekvencu (&0D &0A) Kada brojač u FOR petlji dostigne vrednost 296 štampač u stvari dobija 300-ti bajt jer je u međuvremenu računar poslao dva puta po &0D &0A tako da izlazi iz grafičkog moda i sledeća (zadnja) četiri primljena bajta štampa prema ASCII tabeli znakova tj lapiše .[[[[Ova dva napoželjna dejstva se jednostavno sprečavaju se POKE #8804 0 (eliminište konverziju kodova od &A0 do &AF) i POKE &AC09 &FF sprečava slanja napoželjnog CR/LF

Ako sada startujete bajzik program, trebalo bi da dobijete korektan lapis, slika 2.

Na kraju napomena čitaocima koji nemaju Računara 15/16, III program AMSWGEN (pda se nalazi uputstvo o prepravci kompjutera) Interfeja je hardverski all izmena na prethodnja nikakav problem Potrebno je izvod broj 8 (konaktora za priniar sečivom odvojiti od mase i izolovanom žicom povezati sa pinom broj 12 IC-a 8255 — i izmena je gotove

Od labela **busypr** je Zaharov program koji se poslu učitavanja inicijalizuja se POKE &BDF-2 0 POKE &BDF3 &A0, a takođe traba otkucati i gornja dva POKE-a Oni koji na poznaju mašinsko programiranje a desadilo im ja da posle svakog učitavanja inicijalizuju program se četiri POKE-a neka unsou ceo listing i tako asemblriaju Nakon učitavanja program se inicijalizuje samo se CALL <ORG-adresa>

Ivan Randelović

Bajtovi lične prirode

Rokovi za predaju rukopisa su takvi da ovaj tekst pripremamo pre nego što su „Računari 40 uopšte izašli iz štampe, pa je još rano za reakcija čitalaca na našu novu rubriku Nadamo se međutim, da ćemo već za mesec dana biti u prilici da predstavimo i prve priloge naših čitalaca!

Pošto su „Bajtovi lične prirode nova rubrika, dopustićemo sebi kratko ponavljanja njena koncepcija Koliko vam se puta desilo da potrošite minute III časove svog dragocenog vremena da biste rešili neki naoko sitan problem? I koliko ste puta posle višemesečnog III čak višegodišnjeg korišćenja neke tehnika ustanovili da se iste stvar mogla uraditi mnogo brže lakše i uopšte racionalnije? Ako je problem namučio vse varovatno muči i mnoge druge ako vse je rešenja oduševilo, možda će posredstvom naše berze ideje trikova i kratkih zapažanja oduševiti još nekoga! Ova rubrika nije ograničena samo na programarske ideja — bavićemo se operativnim sistemima aplikativnim programima i uopšte bilo kakvim softverskim paketima koji su dovoljno složeni da u sebi sekriju neku ozbiljniju tajnu Nismo, osim toga ograničeni ni na softver — dobrodošao je svaki hardveraki trik koji se može izložiti u par redova

Posebno bismo želeli da jednom publikovani bajtov na padnu u zaborav — objavljivaćemo komentara, ispravke i dopuna koja pošalju ostali čitaoci Posebno su interesantne ispravke i dodatna objašnjenja — ukoliko se pokaže da je neka objavljena interpretacije bila pogrešna značaj ove priloga je porastao pogrešna interpretacija nas je dovela do tačna III bar tačnije! Ponekad ćemo uz sva to objavljivati i pitanja to jest probleme koji nas muče all koja ne možemo da rešimo, možda ih je neko već rešio!

Sistem obeležavanja priloga je veoma značajan za ovakve rubrika Odlučili amo da „bajtova' obeležavamo rednim brojevima i da u srednjim zagradama pominjamo temu i računar Pri toma koristimo skraćenica iz naleg , indekak tema (pogledajte , Računara 22 III „Računara 33) koja su manje-više očigledna PJ su programski jezici OS operativni sistemi PC je IBM PC i toma slično Nomenklaturu ćemo praciniierdefinisati i učiniti savršenijom kada rubrika zaživi i kada pravci njanog razvoja postanu jasniji.

1B [OS MSDOS, PC] Koliko ste puta otkucali COPY * . DEL * . III nešto slično? Niste mirali da kucate zvezdice — posao sbevlja I **COPY** odnosno **DEL** .. Pažnja! DEL briše sva datoteka u tekućem a DEL .. u *osnovnom* katalogu bez obzira gde se nalazili

11 [STAM YUSET, FX85] Ugrađnja NLQ YU slova u (relativnoj nova Epsonove štampače FX—85 I FX—105 je prilično problematična — oblici su doduše, smešteni na očekivano mseto ((EP) ROM 2764 na ekspanzionoj kartici) all je kodiranja znakova veoma kompilkovano Sva slova pra svega na zauzimaju jednak prostor znak čiji je kod &20 (blanko) počinje od &526 gde je upisena njegova širina. Lokacija sledećeg znaka se računa po formuli SLEDEĆI = PRETHODNI +2-ŠIRINA+B pošto se na &526 nalazi nula sledeći znak (uzvičnik) počinja od &3A2E Svaki znak se sastoji od dva potovina (avaka zauzima ŠIRINA+4 bajta) koja slava lapisuja u avakom od prolaza matrica je takva de velika slova treba deformisati da bi im se dodale kvakica što na primar nija slučaj na Canonu PW—1080A Slova koja imaju *descender* su nešto drugačije kodirana — prve bajtova druge polovina znaka treba prevrnuti Program za analizu i promenu seta znakova objavićemo u nakom od sledećih brojeva „Računara

12 [STAM YUSET, PANASONIC]: U vezi se prethodnim — da li je neko od čitalaca Računara uspeo da ugradi YU znaka (*makar i drafti*) u Panasonicove štampače?

13 [OS MSDOS, PC] I ima direktorijuma može da ima ekstenziju — slobodno otkucajte MKDIR PROBA DIR III čak MKDIR PROBA.BAT Direktorijumima je zgodna pridružiti akstenziju DIR i docnije kucati nešto poput DIR* DIR III u skladu se bajtom 10 DIR DIR.

14 [PROM] Kada u neki od pomenutih štampače III sličnih uređaja traba ugraditi nov EPROM najprirodnija je odlučiti se za tu odgovarajućeg kapaciteta Ukoliko vam međutim 2764 nedostaje poželjeća da upotrabite 27128 i stvar neće raditi A mogla je — u 27128 trabe upisati dva identična bloka od po 8 kilobajta!

15 [IGRE TETRIS, PC] Tetris je gotovo nemoguća igrati na brzim AT kompatibilnim računarima — figurise prabrzo padaju! Ako je već računar a pramiljan prakidačem za promanu takvancija startujte Tetris odaberite pati nivo i *kada igra počne*, uspostre računar igra će i dalje biti prabrza kada stignete do devetog nivoa all bar možete da sakupite pristojnih 8000 poena.

16 [OTXT WPERP.4, PC]: Bajt broj 4 iz prošlih Računara se ne odnosi na WordPerfect 4 1 — očito se radi o begu koji su programeri „skrivili" pripramljajući novu verziju.

17 [PJ PASCAL, PC] Suprotno onome što tvrdi HELP, za rad se grafikom *Turbo Pascalu 4.0* treba priključiti i modul GRAPH TPU Kada je već prenesete na hard diak, neće biti zgoreg da pogledate i demonastracioni program GRDEMO PAS pri čemu se na diaku mora naći i draiver za odgovarajuću karticu na primer HERC BGI.

Prilje se za vu rubriku šaljete na adresu Računari (za Bajtove iz haosa), Šulevar vojvode Mišića 17 Beograd

Uređuje
Dejan Ristanović

Mali oglasi

SPEKTRUM

Najnoviji tematski kompleti od crtanih do simulacija Već imamo Nordstar Black Lamp Katalog besplatan komplet 1500 din Saša Jakšić Stanislava Sremčevića 5 Beograd tel 011/402-428

* TURBODRIVE *
DISKETNI INTERFACE ZA SPECTRUM
1 TURBODRIVE INTERFACE kompletan ... 300 000 d
2. FLOPPY DRIVE 360k/1eM kompatibilan ... 420 000 d
KOMPLET 1 i 2. ... 650 000 d
3. POJEDINAČNI DIJELOVI ZA TURBODRIVE
— FDC WD1770 ... 80 000 d
— s.pločica (nemetal iz. rupe) ... 25 000 d
— s. pločica (metal iz. rupe) 60 000 d
— TTL čipovi komplet ... 60 000 d
— EPROM programiran. ... 30 000 d
— ZX rubni konektor ... 19 000 d
— FLOPPY konektor 34 p ... 19.000 d
— CENTRONICS konektor 36 p ... 16 000 d
— D konektor 9 p ... 4 000 d
— FLOPPY kabel 34 p (1 m) 9 000 d
— podnožja precizna (po pinu) 90 d
OSTALI HARDWARE ZA SPECTRUM
— CENTRONICS INTERFACE ... 90 000 d
— EPROMER sa DC/DC konvertorom 9/25 V ... 300 000 d
— DC/DC konvertor 9/25V 30 000 d
— Kontroler za koračne motore i 2 koračna motora (7 5 st. 13 Ncm) ... 280 000 d
— AD konvertor 8 kanala vrijeme konverzije 100 mikrosec 200 000 d
— rezervni dijelovi — ULA ... 75.000 d
— ROM ... 36.000 d
— 4116 ... 8 000 d
— 4164 ... 7 000 d
— proširenje RAM-a na 48k ili 80k ... 60.000 d
— folije ... 45 000 d
— folije za PLUS ... 60 000 d
SERVICE ZA RAČUNARE SPECTRUM I COMMODORE C-64 C+4, C-16/116 C-128
HARDWARE SERVICE P P 96 42300 ČAKOVEC tel (042)54-795

KOMODOR

C-64 EPROM moduli kablovi programi na disku i kazeti uputstva. Davor Borošak. Froudeova 68 41000 Zagreb tel 041/522-508

Najnoviji kompleti za C-64 bez Intro i demo programa Memorijski snimak!!! K1 STRIP POKER 2+ HERCULES 1-2 OVE the TOP SHOT EM'UP CONSTRUCTION K K2 KARNOV 1-9 OXFORD P DELTHAPL K3 SPARTACUS BEACH BUGGY G.A.C. K4 Najnoviji hitovi Provjerite! Jedan komplet sadrži 20-ak igara i zauzima stranu C-60 kasete tako da morate za napuniti kasetu naručiti dva kompleta. 1 komplet 1000 kaseta 1500. Pojedinačno 150 Prva 2 naručioca dobivaju QUICKSHOT 2+!!! SANJIN FRLAN Zagrebačka 18 52220 LABIN tel 052/852-610

tel 052/852-610 PSL HARDWARE & SOFTWWARE Ltd Razdjelnik 7500 Komplet 1000 strana DISKA. 500 2D disketa 2100 Pretplata na jednu kasetu 3000 Najnoviji disk prg KARNOV (1D) BARD S TALE 3(4D) PIRAT DISC S PACK (4D) SPARTACUS (1D) SANJIN & NENAD — LABIN — C-64

— SORILAČKI — SVEMIRSKI — USLUŽNI — VJEČNE igre — PORNO — POKE igre Jedan komplet sadrži 20—30 prg i zauzima stranu C-60 kasete i kom 1000 d kaseta 1500 d tel 052852610 SANJIN FRLAN Zagrebačka 18 52220 NENAD STEMBERGA, Zagrebačka 14 LABIN

Od filmskih do sportskih kompleta i hit komplet ,LETO 88 i Road Warrior 2, Xenon Poltergeist, Tough Guys Time Magic 35 programa sa uputstvima 2500 dinara +kaseta+PTT Miodrag Vojković Bulevar revolucije 111/I, Beograd, tel 011/403-852

INTERNATIONAL STUDIO — komplet (30 prg.)+kaseta C60+uputstva+PTT=5000 din. Dva kompleta=7000 din Nema demoa i intros. K5 Rimrunner Winter Olympiad 2 Macro Assembler Arkanoid 2+ K6 Street S Soccer IQ Samurai W + Boxing 1-3 Besplatan katalog Posjedujemo veliki broj intro i demo makera. Stari kupci imaju popust 20% Vlastimir Kojović XI Krajiške divizije 58 75000 Banja Luka tel 078/80-854

Garfield Computer Club vam i ovog meseca nudi najnovije programe za vaš C64 Snimamo na kazetame (komplet + kazeta + PTT = 6000 din) i vašim disketama BESPLATAN KATALOG G C C (za C 64), A. Pomoraca 23, 41020 Zagreb tel 041/527-699

COMMODORE 64 & 128
HARDWARE SOFTWARE LITERATURA
34000 KRAGUJEVAC

Ne verujem da neko nudi više programa za VC 20 C 16, C+4, C65, C128 i Amigu Tražite katalog Đerman Šandor 23000 Zrenjanin Rade Končara 23 tel 023/30-969

C-64/128/CPM/AMIGA 500 Prodajem uslužne disk programe i disk igre Besplatni katalozi Radovan Fijember Klaićeva 44 Zagreb tel 041/572-355

ELECTRIC SPY!! Igre uslužni programi. Komplet 1500 dinara. Besplatan katalog. Electric Spy Marička 13, 16000 Šabac, tel 015/25-597

KODAKSOFT nudi najnovije programe za Commodore 64/128 Niske cijene Auto Irke, korlanički programi, sec programi muzički programi i kompletirani programi Tel 072/611-308

ATARI

Atarijevci žele li da koristite literaturu (Atari USER knjige) a da je ne kupujete Javite se! Tel 062/74-388

Atari Master Club XL/XE. Za vaš Atari XL/XE najveći izbor programa literature uputstava. Kursevi za učenje bejzika na kasetama Atari User Super popusti ekspres usluga i umerene cene. Za katalog pošalji 500 dinara Slobodan Jovanov Prvomajska 2-a, 23000 Zrenjanin

Za Atari XL/XE najbolji kasetni hitovi. Kvalitet zagarantovan Naručite besplatan katalog!! Goran Veljakov Sava Kovačević 70-2/8, 91000 Skopje

Prodajem Atari 520 ST sa jednostranom disk jedinicom i monitorom 124 Jovan Anđelković tel 011/687-953

Najnoviji programi za Atari 500 XL 130 XE u normalnoj i turbo brzini Miroslav Mikić Đure Salaja 41 42000 Varaždin tel 042/53-977

AMSTRAD

Amstradovci! Highlight crew (staro ime Micro club) nakon duže pauze ponovo je sa vama! Svi ljetni hitovi za Schneider (i Amigu) u kompletima pojedinačno na jednom mjestu Cijena jednog kompleta 2500 din.+kvalitetna C60 kazeta (2000 din) ili TDK C60 (5000 din)+PTT
Programe možete dobiti i pojedinačno na kazeti ili disketi Komplet 44 Rolling thunder Stiffip & Co Return to Oz Worm in paradise Professional tennis. Derwish Komplet 43 Platoon 1—3 Gryzor 1-3 Gunsmoke 1-5 Andy capp Slaine Warcars Komplet 42 Anarchy Tir na nog Galactic games 1-5 Rolling thunder Academy Rebel planet
Komplet 41 Mask Mask II Sepuleri Mentor Pink Panther Baseball Predator 1—4 Tetris Battleships Boubble bobble 1—2 Komplet 40 Stadium Beyracer Warhawk Knightmare Rampage Joe Blade Jetbike simulator 1—6. Side arms Star wars Rally driver
Komplet 39 Dan Dare II Madballs Garfield Flying shark Guadal canal 1—3 Agent X II 1—3 Arkanoid II Red LED Froggy Arkham manor Komplet 38 Bobsleigh Buggy boy Naja Xor Flash Gordon 1–3 Catch 23 Boulderdash II Dustin Komplet 37 Moles Ramparts Ransom Strike Match day II Microball Killering Zora Komplet 37 Ini Karate+ Uridium Deflector Paperboy 2. Milk race Phantom club Masters of the universe
ne 5 naručenih dobijate još dva kompleta besplatno
Disc program Art studio Mini office II Masterfile III Turbo pascal 3.0 Tasword 6128 Wordstar 3.34 YU Cijene 5000 —5000 din Svaki zauzima jednu stranu diskete Posebno Amx pagemakera +neophodna uputstva (110 stranica) — 20000 din +disketa. Za katalog pošalji 500 din u pismu
Highlight crew Branimir Jeranko Lenjingradska 72 41000 Zagreb

DANSOFT Amstrad 464 predstavlja programe.
— Titlemaker+— napravite vlastite verzije igara
— Change 2.0 — upisujte svoje poruke u omiljene igre
Programi+kaseta+ PTT + uputstva=7900 din! Kvalitet zagarantovan A. Duniskog 17 16000 Leskovac. tel 016/43-710

Electric Spy!! Direktno iz kompjutera. Komplet 15000 dinara Besplatan katalog Electric Spy Mišar 168 15000 Šabac tel 015/23-410

STAR WARS
nudi: bedlam,
crazy cars,
andy capp,
driller,
anarchy...
call me !!
Siniša Slobode Đakovo 054/842-460

AMIGA

Amiga — Highlight crew vam predstavlja najnovije programe za vašu prijateljicu Deluxe Special (DPAINT III III) Compacted disc #11 (9 igara) Vortex 42 Utility disc #3 Racket ranger Buggy boy Sentinel F-18 Interceptor Battleships II Zero gravity International football World tour golf Barbarian II i još 400 ostalih programa za profesionalce i djecu od 5 do 55 godine i ostale P.S. Specijalno smo za vas pripremili 20-ak disketa sa 2—5 igara na jednoj! Naručite besplatan katalog na adresu Branimir Jeranko Lenjingradska 72 41000 Zagreb i provjerite zašto smo najbolji!

IBM

NAJVEĆI IZBOR NAJNIŽE CIJENE SOFTWAR-a za IBM PC 000 najnovijih uslužnih programa i 150 igara PCAD v 2.0 kompl ORACLE za XT i AT QUATTRO WORDSTAR 2000 Graphic MS C 5.0 SYMPHONY 2.0 itd Snimam prog na diskete 5.25 i 3,50 Tjedno novi programi tražite nove besplatne kataloge Zdenko Bakas Ivana Milutinovića 34 41040 Zagreb Tel (041/254-581

EE IBM PC SOFTWARE

Najveći izbor SOFTWARE-a za IBM PC u Jugoslaviji po najnižim cijenama. COPY II PC v5.01 WORDPERFECT v5.0 VACCINE COMPUTER VIRUS, TURBO C v1.5 BOEING GRAPH v4.0 HARDWARD PFS PUBLISHER MICROCAP II v3.7 MULTIPLAN v3.0 Igre TEST DRIVE, BOULDER DASH i još preko 307000 Kb vrhunskih programa najpoznatijih svjetskih proizvođača Literatura! POKLONI! EKSTRA popusti! Katalog besplatan! Rok isporuke 24 sata! EE Software Martićeva 31 75000 Banja Luka tel 078/40-940

Charlie Soft

Novo za IBM PC! Kompleti!, Program+uputstvo+diskete+plastične konce. Kada kupujete kupujte kvalitetno i kompletno! Otes B-35 ul 5/7 71210 Ilidža Tel 071/628-519

KUPUJEM program za optičku rekogniciju karaktera za IBM PC sličan Recognita -I Marjan Janev Bul Jane Sandanski 36-3/2 Skopje tel 413-813, posle 15 časti

Literatura za IBM PC Wordperfect 5 0 Chi Writer 2.56 Turbo Prolog Toolbox MS Cobol MS Fortran 4 11 Turbo C i dr Dejan tel 011/150-835

LakoSoft programi BEZ IKAKVE OBUKE ZA PC! Početnici oblikovaćete PC pomoću reči govornog jezika! Mali privrednici PC-em pratićete poslovanje kako zatražite s izuzetno lako! Za prospekte pozovite 011/138-072 457-072 (Boža) ili pišite na LakoSoft Alekse Nenadovića 12—14 11000 Beograd

RAZNO

TIM 011
pouzećem prodajem
— komplet otpornike i kondenzatora (5800 dinara)
— komplet delova za ispravljač bez štampane pločice i transformatora (84000 dinara)
Zoran Đurišić Ranke Tajsića 5 11000 Beograd

OPRAVKA računara „Spektrum", „komodor" i atari Proširenje memorije za „atari ST" Veliki izbor uslužnih programa i igara za atari ST po povoljnim cenama racionalno snimanje — nema Introa i sličnog Petar Putnik Bare Vlajkova 86 21220 Bečej

Pozivalac odlučuje koji će od dve paralelna telefona zvoniti posredstvom elektronskog sklopa koji je instaliran kod pozvanog Uputstvo za izradu sa štampanim pločicama Tel 011/666-675

Popravka i nabavka IC kola i štampanih pločica za računar Galaksiju tel 011/723-313

HARDVER

Prodajem
— monitor Schneider GT 65 za CPC 464
— modem Dataphon S 21 D sa interfejsom za C-64
Informacije na telefon 722-740

Prodajem Schneider CPC 8128 i kolor monitor CTM 644 tel 041/677-610

Hitno i veoma povoljno prodajem CPC-464 sa zelenim monitorom! Vladan Matić, 1 L. Ribara 36 11318 Miloševac

Prodajem Atari 800XL, originalni kasetofon i programa Tel 011/189-068

Prodajem PC AT Kompatible Floppy 1 2 Mb hard 30 Mb herkules monochrom monitor Tel 022/71-205

631

BIGZ VAS POZIVA NA PRETPLATU PO TRECOJ NAJPOVOLJNIJOJ PRETPLATNOJ CENI

BOJ NA KOSOVU 1389—1989.

u tri velika toma

BIGZ-ova jubilarna trilogija ilustrovana, u luksuznoj opremi:

1. KRALJ SRPSKOG CARSTVA
2. LAZAR HREBELJANOVIĆ — istorija, kult, predanje
3. JUNACI KOSOVSKE LEGENDE

Trilogiju je napisao prof. dr Rade Mihaljčić

Obim kompleta oko 1100 strane velikog formata 19,2×29,5 cm, 140 ilustracija, knjige su štampane ćirilicom u tvrdom povezu, sa zaštitnim omotom, komplet je u zaštitnoj umetnički oblikovanoj kutiji

— **80.000., dinara**, plaćanje odjednom, samo za članove Klube čitalaca BIGZ-a
96 000, dinara, plaćanje odjednom, za ostale
120.000,. dinara, ako se plaća u ratama
Isporuka kompleta urednim pretplatnicima je po izlasku iz štampe i po uplati celokupnog iznosa pretplate Knjige izlaze iz štampe u januaru 1989 godine

Za pretplatnike specijalan poklon: luksuzna olovka u zaštitnoj i umetnički oblikovanoj kutiji

2. Dragoljub Živojinović: KRALJ PETAR I KARAĐORĐEVIĆ

Izlazi krajem oktobra 1988
— **29 400 dinara** odjednom samo za članove Kluba čitalaca
— **33 600, dinara** odjednom za sve ostale
— **42 000 dinara** ako se plaća u ratama

3. Vladimir Jovanović: USPOMENE

Priredio Vasilije Krestić. Obuhvataju period od hajdukovanja harambaše Ostoje Spuže do balkanskih ratova. Oko 500 strana velikog formata oko 70 fotografije i faksimila tvrd povez sa zaštitnim omotom u boji ćirilica izlazi u oktobru 1988
— **28 000. dinara** plaćanje odjednom samo za članove Kluba čitalaca
— **30.400. dinara** plaćanje odjednom za sve ostale
— **38.000,. dinara** ako se plaća u ratama

4. Edicija HITAC, prvo kolo u 5 knjiga

1 **Džozef Hejvud BERKUT** politički triler 2 **Bret Isten Elis MANJE OD NULE** roman o mladoj generaciji 3. **Čarls Bukovski MUZIKA VRELE VODE** priče 4 **Stiven King KERI** horor roman 5 **Henri Miler OPUS PISTORUM** erotski roman Oko 1 500 strana format 12×20 cm broširani povez komplet u zaštitnoj kutiji latinica izlazi krajem oktobra 1988

— **70 000 dinara** plaćanje odjednom samo za članove Kluba čitalaca
— **80 000, dinara**, plaćanje odjednom za sve ostale
— **100 000 dinara** ako se plaća u ratama

UČLANITE SE U KLUB ČITALACA BIGZ-a — NABAVLJAĆETE KNJIGE SVIH JUGOSLOVENSKIH IZDAVAČA PO NAJPOVOLJNIJIM USLOVIMA!
Tel 650-235, 653-899, 653-763

(Mesto i datum)

BEOGRADSKI IZDAVAČKO-GRAFIČKI ZAVOD
11000 BEOGRAD Bul vojvode Mišića 17
pošt fah 340 tel 653-763, 650-235 653-899

I

Pretplaćujem se na knjigu BOJ NA KOSOVU komplet u knjige po drugoj pretplatnoj ceni od

86.000 ako se plaća odjednom
80.000 samo za članove Kluba čitalaca
120 000 ako se plaća u ratama

II

Pretplaćujem se na sledeća BIGZ-ova izdanja ____________

navesti redne brojeve iz ovog oglasa

Ukupan iznos od ____________ dinara plaćam
a) Odjednom po specijalnoj ceni za članove Kluba čitalaca BIGZ-a a na osnovu članske karte Kluba br ________ u roku od 8 dana po prijemu BIGZ-ove uplatnice
b) odjednom po ceni za sve ostale u roku od 8 dana po prijemu BIGZ-ove uplatnice.
c) u ____________ mesečnih rata (najmanji iznos rate 10 000 dinara najviše 8 rata) uvek uplatnicama koje su dobili od BIGZ-a
Isporuka knjige po izlasku iz štampe po uplati celokupnog iznosa pretplate U slučaju spora nadležan je odgovarajući sud u Beogradu

(Prezime ime oca i ime)

(Zanimanje) (Telefon u stanu — na poslu)

(Adresa stana broj pošte mesto ulica i broj)

(Organizacija u kojoj je pretplatnik zaposlen i njena adresa)

(Potpis pretplatnika broj lične karte i mesto izdavanja)

Overa o zaposlenju penzioneri prilažu pretposlednji ček penzije (obavezno samo kod plaćanja knjige u ratama)

HELP



U ovom broju kao i u prethodnom rubrika „HELP" je prilično smanjena u odnosu na svoj uobičajeni obim. Letnji meseci očigledno utiču negativno na znatiželju vezanu za računara. Sudeći po pristiglim pitanjima korisnici Amstradovih računara imaju dosta problema sa kompajlerima ali i veliku želju za ozbiljnim programiranjem. Ova će rubrika nastojati da pruži najveću moguću pomoć takvim korisnicima u njihovim prvim i najtežim koracima ka ozbiljnom programiranju. To se, naravno, odnosi i na korisnike svih ostalih računara.

Kratak pregled carinskih propisa je namerno uvršćen u ovaj broj zbog toga što jedan broj vlasnika računara koristi letnja praznike za putovanja u inostranstvo, odakle se uvek vrate i sa nešto opreme za svoje kućna ljubimce. Da pri povratku ne bi bilo naugodnih iznenađenja na granici, pažljivo pročitajte šta i kako možete uvesti i koliko to iznosi u čvrstoj valuti.

Srećno Programiranje

Tražimo — nalazimo

Zlatka Tomić iz Beograda ima ploču/tastaturu za galaksiju, koja bi rado ustupio nekom čitaocu „Računara. Takoda ima i pravod uputstva za HP-41, što će sigurno obradovati **Dragana Despetevića**. Adresa Zlatka Tomića je Skojevska 41/18 11000 Beograd.

Marko Hudiček iz Novog Kneževca ima ORIK-64. I moli da mu se jave oni čitaoci „Računara koji imaju shema hardverskih sklopova koji bi se mogli priključiti na orika. Posebno ga zanima interfejs za džojstik.

Bajt u boci

Nenad Dobrijević iz Vibovca je došao u posed izvesna crna kutija sa dva kabla koji vire iz nje i traži informaciju o tome čemu ona služi.

Posiati ste suviše malo podataka da bi se bilo šta pouzdano moglo reći. Kada biste obezbedili fotografiju ili dobar crtež spoljašnjosti i unutrašnjosti kutije, stvari bi verovatno bile mnogo jasnije. Podaci o tome kako ste došli do te kutije, postoji li unutra neki transformator, postoje li vodovi za napajanje itd. mogli bi takođe da budu dragoceni u otkrivanju njiana namena.

Uvoz — Carinski propisi

Emil Kopjar iz Varaždina želi da zna kolika su uvozna kvota za računara i periferalija.

Pri svakom ulasku domaći putnik može uneti bez plaćanja carina robu u vrednosti do 20 000 novih dinara a državljani SFRJ na privremanom radu u Inostranstvu do 100 000 dinara. Obračun ovih kvota se vrši po tekućem deviznom kursu.

Naši i strani građani imaju pravo da pri svakom ulasku u Jugoslaviju unesu predmete u vrednosti do 175 000 novih dinara. Tolika je i vrednost pojedinačnih pošiljki koja mogu primati iz inostranstva (poštom avionom vozom itd.) Osim toga imaju pravo da jednom godišnja uvezu računarsku opremu u vrednosti do 300 000 novih dinara. Građani na privremenom radu u inostranstvu mogu jednom godišnja uvesti predmete u vrednosti do 525 000 novih dinara. Za sve ovo se naravno plaća carina. Kvote se obračunavaju po statističkom deviznom kursu a carinska osnovica po tekućem deviznom kursu.

Po ovogodišnjem statističkom deviznom kursu 175 000 novih dinara je 183 USD 75 LSTG, 225 DEM a 300 000 novih dinara je 633 USD 343 LSTG 1030 DEM. Procena vrednosti robe je naravno stvar carinskog službenika a iskustvo govori da se uz nešto sreće uvozna kvota za računar može sasvim pristojno „rastegnuti".

BBC — Muzički programi

Saša Stojkovski iz Zagreba traži informaciju o muzičkim programima za BBC.

Među kvalitetnije muzičke programa spadaju MUSIC SYSTEM firme Island Logic i MUROM firma Beebugsoft. Prvi se prodaja na diskati i košta 50—60 LSTG a drugi u ROM-u i košta 35 LSTG. Za detaljnije informacija možete se obratiti beogradskom klubu programera (Dom Omladina Beograda za Klub programera Makedonska 22/5 11000 Beograd).

Dejan Ristanović

Amstrad 6128 — Kompajliranje fortran programa

Primož Vugrinc iz Maribora ima FORTRAN 80 kompajler na „amstradu 6128" koji radi pod CP/M 3.0 operativnim sistemom. Njegov problem je u tome što nema podataka o tome kako da kompajlira program.

Izvorni kod programa pripremite u bilo kom tekst procesoru u non-document (ASCII) modu sa ekstenzijom FOR. Učitajte CP/M i otipkajte F80 čima startujete kompajler. Na prompt (*) otipkajte =Filenama gde je Filenama naziv fajla sa izvornim kodom (i ekstenzijom FOR koja se na navedi u Filename). Po unošenju navedana komande vrši se sintaksna analiza i prijavljuju se greška. Ukoliko nija bilo greška otipkajte **Filenama, Filenama**, čima se izvršave sam proces pravođenja u toku koga se atvara Filenama REL fajl.

Po uspešnom pravođenju izadite u CP/M sa CTRLC i otipkajte L80 čima se startuje linkar. Na prompt (*) odgovorite se **Filenama, Filenama /N** čima otpočinje proces linkovanja. Na sledeći prompt odgovorite sa FORLIB/S a na onaj posla njega se /E čima se linkovanja završava.

Sada na disku imate FOR, REL i COM fajlove od kojih prvi sadrži izvorni kod drugi sadrži objektni kod (koji možete izbrisati ili apremiti u naku biblioteku potprograma) a treći sadrži izvršni kod koji se izvršave navođenjam imana Filenama.

Dejan Predić

Štampači — MPS-1200.

O Jančevski (ime je nečitko napisano) traži nakoliko dodatnih informacija o printeru MPS-1200 (prikazanom u RA 36).

Kod MPS-1200 interfejs se nalazi u kertridžu tako da je njegova izmana krajnje jednostavna. Pri promani interfejsa merate kupiti i nov kabl koji odgovara interfejsu.

Firma KONIM vam može prodati Centronics interfejs za MPS-1200. Traba naravno pre poručivanja da sa raspitate da li ga imaju na skladištu.

Uz odgovarajući raspored DIP-prekidača MPS-1200 postaje potpuno FX-kompatibilan što znači da odlično razuma EPSON-komande koja mu naki program sa C-64 šalja.

MPS-1200 ima mogućnost definisanje korisničkih karaktera. Da li će to biti YU-slova ili nešto drugo atvar je korisnikove želja i potrabe.

Ovaj printer podržava visoku rezoluciju. Sposoban je da štampa ček i slika sa „amiga".

MPS-1200 je po svejim karakteristikama praktično identičan CITIZEN-120 štampaču.

Voja Gašić

C-64 — Ukjučivanja VIC-a

Dario Sušanj iz Zagraba želi da drži ukjučen VIC za vreme učitavanja programa kako mu ekran ne bi zvijao prazan.

Rutina za učitavanje programa se nalazi u ROM-u pa njana izmana (a to je neophodno da bi VIC ostao ukjučen) zahteve izmenu ROM-a, što je ipak malo preskupa cena slika na ekranu.

Možete, naravno, napisati svoj program za učitavanja ili prepisati onaj iz ROM-a pa ga srediti tako da ostavlja sliku na ekranu. Nije li ipak, bolja da koristite neki od standardnih programa za brzo učitavanja?

Vladimir Stakić

Casio FX-702p — Perifarijska oprema

Bokun Ratko iz Hadžića želi da zna kakvu perifernu opremu može nabaviti za CASIO FX-702P.

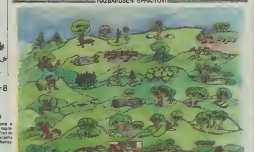
Budući da već imate kasetni interfejs FA-2 jedini periferni uređaj koji još možete nabaviti za svoju mašinu je termički printer FA-10 koji štampa ne vakom papiru 20 znakova u redu brzina mu ja 1 red u sekundi. nema mogućnost definisanje korisničkih znakova, niti mogućnost štampanja grafika.

Ukoliko vam traba mašina koja može da „vozi više periferija (disk modem RS-232, ploter, marne instrumente itd.) jedino rešenja je da nabavite HP-71B.

Vladan Dugarić

Razbarušeni sprajtovi

Uređuje Slobodan Perović



1+1+1+ . . .=8

Igre rađene po „crtićima"

YABBA DABBA DOO

Nakon uspjeha crtanih filmova o porodici Kremenko i Kamenko napravljena je i ova simpatična igra. Fred se mora udvarati Vilmi i u isto vrijeme sagraditi najljepšu kuću u gradu. Nemojte očekivati lagan posao.

SCOOBY DOO

Kao pas iz crtića Hanna i Barbera treba da nađete i oslobodite četiri prijatelja koje je zarobio ludi naučnik. Napadaju vas razna smetala ali ni Scooby nije bespomoćan — udara pesnicom kao Mike Tyson.

ROAD RUNNER

U ovoj igri glavni likovi su ptica trkačica i njen veliki neprijatelj kojot. Vi ste naravno ptica trkačica a kojot vas na 16 nivoa i bezbroj načina pokušava spriječiti u trčanju.

THUNDERCATS

Mračni Mum-Ra je ukrao Oko eldera i vi kao Liono ste zaduženi da ga vratite. Naoružani čudotvornim mačem probijate se kroz dvadeset nivoa i lijepo se zabavljate.

YOGI BEAR

Medvjed Yogi treba da oslobodi svog sinčića Boo-Boo-a koji je zarobljen u pećini Jeloustonskog parka. To se mora učiniti prije dolaska zime i neizbježnog zimskog sna a ometaju ga stanovnici i čuvar parka.

MASK II

Igra prati pobjede Mete Tragača i njegove grupe „Mask" u neprekidnoj borbi između Zemlje i protivnika koji se skrivaju pod imenom „Venom". Met je teško naoružan i kotrlja se okolo u „Olujnom sokolu" — drumskoj krstarici koje je stalna meta protivnika.

GARFIELD

Mačak Garfield je u velikoj nevolji. Njegovu djevojku (macu) Arlena uhvatio je šinter i Garfield mora da je spasi. Odlična grafika i animacija daju draž crtanog filma ovoj simpatičnoj igri.

BASIL

To je ime čuvenog miša-detektiva koji živi u podrumu kuće Šerloka Holmsa. Zadatak je da oslobodi Dr Dosona koga je zatočio zli pacov Retigan. Igra je namijenjena djeci ali ni odraslima nije zabranjen pristup.

Mario Peter

WINIE THE POOH

in the bundred acre wood

Vini Pu je simpatičan medvedić iz Diznijevih crtanih filmova. On obožava da jede med da se igra i da speva. Ume čak i da radi ništa (do nothing). Okružen je mnogobrojnim drugovima. Tu su i prase Piglet Vinijev najbolji drug, zatim stari Zec koji ima jako lepu bašlu i bogatu setavu. Ejor tužni magarac po kome nikad ništa nije u redu. Kenga brižna i pedantna majka kengur i njena beba Ru koji voli da se igra u pesku. Sova koji je veoma mudar i obrazovan i konačno Tigar koji strašno voli da jurca.

Problem je nastao kada se jednog jutra Vini Pu probudio i zaključio da mu nedostaju neke stvari. U početku je mislio da ih je prosto zaturio negde pa kad ne može da ih nađe. Međutim, posetio ga je Piglet žaleći se da i njemu nedostaje nekoliko stvari. Tu ih je rad prekinule. Posle ručka su otišli da šetaju šumom gde su sreli Zeca i Sova kojima je takođe nestalo par stvari. Konačno naleteli su na Kristofera Robina koji im je sve objasnio. Prošle noći je duvao jak vetar i pomešao sve u šumi. Svakom stanovniku šume nedostaje ponešto (izuzev tigru). Vaš zadatak je da izgubljene predmete vratite vlasnicima a neke predmete treba ostaviti na određenom mestu. U svakoj igri treba vratiti deset predmeta od ukupno 40 pa vam neće biti dosadno i sada budete igrali više puta.

Ekran je podeljen na dva dela, gornji u kome je ispis lokacije i predmeta (ako na toj lokaciji postoji predmet) i donja četiri reda koja su rezervisana za poruke i komande. U ovoj igri ne treba pisati nikakve naredbe. Kada stignete na lokaciju kompjuter vas obaveštava da li ima predmeta na njoj. Ako ime pritisnite bilo koju tipku i dobićete opis lokacije koji je vrlo kratak. Opet pritisnite nešto i u gornjem delu se pojavljuju komande (tj. radnje koje na toj lokaciji možete da vršite) a u donjem delu birate pravac u koma želite da se krećete ili uzimate (tj. ispuštate) predmet.

Komande birate na dva načina: kod „atarija ST" možete ih kliknuti ili pritisnuti početno slovo (npr. 1 — za prvu naredbu, W — za zapad, T — za uzimanje). Ako ste slučajno pritisnuli taster dok ste čitali neku poruku pritiskom na „BACKSPACE" moći ćete ponovo da je pročitate. Taster „ESC" vas vraća na početnu lokaciju (tj. u sobu Kristofera Robina. Ovo je naročito korisno ako želite da iz severnog dela šume brzo dođete u južni. Tu su još i „CONTROL-S" koji uključuje i isključuje zvuk i „C" koji pokazuje šta nosite i koliko predmeta još treba vratiti. Kod verzije za „spi II" računare sve je isto samo umesto klikanja pritiskate „SPACEBAR" i „RETURN" a umesto „BACKSPACE" treba pritisnuti strelicu nalevo. „SPACEBAR" i „RETURN" se koriste i kod „komodora 64" ali kod njega je „F1" umesto bekspejsa, „F3" umesto tastera „ESC", „F5" uključuje i isključuje zvuk a taster „C" zamenjuje „F7".

Svaki predmet pripada određenoj osobi ili mestu. Kada nosite predmet i dođete do osobe za koju mislite da je vlasnik ispustite ga. Ako ste pronašli pravog vlasnika on će vam zahvaliti. Ako niste ništa se neće dogoditi a vi možete pokupiti predmet i nastaviti. U slučaju da niste sigurni kome pripada predmet, donesite ga do Sove i pričajte sa njim a on će vam reći karakteristike vlasnika tog predmeta.

Svaka lokacija može da sadrži samo jedan predmet što znači da ga ne možete ispustiti tamo gde već postoji predmet. Isto tako možete nositi samo jedan predmet. Ako na nekom mestu ugledate predmet i želite da ga uzmete, moraćete da se vratite, ispustite predmet koji nosite, dođete do željenog predmeta i pokupite ga.

Da igra ne bi bila previše laka brinu se uglavnom vremenske nepogode (magla i mist) — sve pokriva tako da ništa ne vidite. Možete jedino da nastavite da hodate a kad se magla razide obrećete se daleko od mesta gde vas je magla uhvatila. Tigar voli da se vija. Dolazi iznenada. Kada dođe pojuriće vas a vi ćete ispustiti predmet koji nosite i naći se malo dalje od mesta gde vas je Tigar propao. Konačno najopasniji neprijatelj je Snažni Vetar (Blustery Wind). Posle izvesnog vremena on će opet početi da duva i premestiće predmete po šumi sve sem onih koje ste vratili vlasnicima.

Srđan Gligorić

Poukova mreža

Spektrum — *Zoran Jovanović*

XECUTOR

POKE 54017,0 bezbroj života za jednog igrača
POKE 53538,0 bezbroj života za dva igrača

BUBBLE BOBBLE

POKE 43871,0 bezbroj života

STAR GLIDER

Konačno jedne prava svemirska pucačina za ... atari ST. Sve ono čega nije bilo u drugim igrama ovog tipa ujedinilo se u „Star Glider"-u. Radnja se dešava u dalekoj budućnosti. Gadni i odvratni uljezi zauzeli su vašu planetu, a vi ste naravno jedini koji je može spasiti!!!!

Nalazite se u svemirskom brodu Leffe po površini planete i imate zadatak da upucavate sve živo i neživo i ono što se miče i ono što se ne miče! Od oružja imate laser kojemu je broj hitaca ograničen jer se pregrijava od silnog pucanja, bombe i dirigovane rakete u kojima se (valjda) nalazi kamera pomoću koje se s lakoćom zabijaju u metu (ne uvek). Brodom upravljate pomoću miša tako da se njegovim micanjem krećete lijevo, desno, gore i dolje, s pritiskom na lijevo uho pucate, dok pritiskom na desno + gore-dolje ubrzavate odnosno usporavate.

Startate iz svoje zemaljske stanice. Na prvi pogled će vas zbuniti svi oni silni neprijatelji, ali će vam se poneki i svidjeti (mislim ne to kako izgledaju, ali ćete ubrzo promijeniti mišljenja).

Pored gore navedenog oružja imate i energetski štit koji vas štiti od određenog broja pogodaka, a onda — Jovo nanovo!!! Imate i gorivo koje se takođe troši. Energiju obnavljate u silosima (stanici) kao i municiju, a gorivo tek ako proletite između dva stupa u pravcu trećega.

Komandna tabla izgleda ovako:

Dolje u sredini je radar, gore su točke pravac u stepenima i vrijednost bodova zadnjeg pogotka. Dolje u sredini koordinator, lijevo gore indikator štita, dolje istrošenost lasera ispod broja raketa koje nosimo, a desno gorivo.

Ukupno ima 36 nivoa. S nivoa na nivo prelaziš kad skupiš 10 000. Najveće zadovoljstvo je kad bacite lov na neprijateljski avion, a onda ispalite raketu na njega.

Željko Manojlović

FIRE POWER

Cilj ove ratne igre je da prije neprijatelja osvojite (osvojite) njegovo utvrđenje (zastavu)! Prvo slijedi biranje tenka. Svaki tenk ima različite svojstva: kao brzinu, manevarske sposobnosti, oklop, zapremninu rezervoara za gorivo.

Ako igrate s prijateljem, svaki igrač zauzima polovicu ekrana. Ako igrate protiv kompjutera, akcija se odvija na lijevoj polovici, a desno je statusno područje. Upravljanje samim tenkom je vrlo lako, mada se problem stvara kad se želite okrenuti za 180 stupnjeva jer možete da zapnete za neku od zgrada.

Startate iz svog utvrđenja. Na putu do protivničkog primjetit ćete neke zgrade koje se mogu uništiti. Ako se desi da uništite svoju zgradu, iz nje će izaći vaši ranjeni drugovi. Na najčešće ih nećete moći ukrcati pa ćete izgubiti nekoliko bodova.

Evo načina kako da najlakše završite igru.

Na početku izaberite tenk SHADOW. Na putu prema protivnikovom utvrđenju postavljajte mine da bi ste ometali kolegu da stigne do vašeg utvrđenja. U igri postoje dva (2) generalštaba, ali u jednom se nalazi zastava. On je na radaru obilježen svjetlucavom točkom boje neprijatelja. Kad stignete do neprijateljskog utvrđenja počnite da pucate po zidovima dok ne napravite rupu. Sad morate da nađete mjesto gdje je zastava, da je osvojite uz stalno ometanje neprijatelja.

Željko Manojlović



REVENGE OF DOH

Nastavak Arkanoida

Reket kojim upravljaš može steći izuzetno velik broj poboljšanja (čak jedanaest). U igri postoje cigle koje se uništavaju nakon prvog pogotka, dok su žute cigle neuništive. Postoje i cigle koje se uništavaju nakon nekoliko pogodaka, ali se trenutak kasnije opet pojavljuju na ekranu. Neke od pogođenih cigli ispuštaju blokove obilježene slovima. Evo šta pojedini blokovi donose:

— B — prebacuje te na sljedeći nivo
— C — lijepi lopticu za reket
— D — deje ti pet loptica istovremeno
— E — razvlači reket s obje strane
— G — reket se ponaša kao duh
— L — reket dobiva razorni laser
— M — deje ti tri loptice istovremeno
— P — dobivaš nagradni život.
— R — reket se smanjuje (zar je ovo poboljšanje?!)
— S — brzina se smanjuje
— T — upravljaš sa dva reketa.

Igra je prilično teška, no olakšaće je vaš unijet POKE za bezbroj života. Mislim da bi ovom igrom konačno trebala završiti era „zidekrša", inače će epidemija pustiti duboke korijene.

Zlatan Hamzić

ROLLERBOARD

U ulozi neustrašivog skejtaša treba preći što više staza (nivoa) koje podsećaju na zapadne skejt parkove. To nije nimalo lako, jer treba preskakati rupe, penjati se uz uzbrdice, paziti na razne korenje.

Ekran je podeljen na dva dela. Gornji gde se odvija igra i donji gde se nalaze podaci o rezultatu, brzini, nivou i preostalom broju života. Za postizanje brzine treba pritiskati dugme, za skok pomeriti ručicu gore, prilikom kočenja ručicu ka vama (dole), a svaki put kad prilazite uzbrdici pomeriti ručicu levo.

Pored svega, na raspolaganju imate i tri vrste skoka koje ćete upotrebiti zavisno od situacije i vaše maštovitosti. To su: obični (istovremeno dugme + ručica gore), sa dvostrukim okretom (istovremeno dugme + ručica levo) i super skok (istovremeno dugme + ručica desno). Svaki skok se različito boduje, ko brzo pređe nivo dobija bonus do 1000 poena. Ukoliko želite da pauzirate igru koristite RUN/STOP taster.

Moje verzije ROLLERBOARD-a krahire nakon startu pa je potrebno primeniti sledeću proceduru. Kad učitate program zaustavite listing. Zapišite na papir broj SYS koji se pojavi u liniji. Resetujte kompjuter sa SYS 64738. Otkucajte SYS koji ste zapisali na papiru, pritisnite RETURN i igra je tu. Ovo isto možete primeniti i kod igre 720 (drugi deo) ukoliko takođe krahira nakon starta.

Zoran Cvetinović

Tetris je pao

Priča se, ne bez razloga, da je ova igra zaludela čitav svet.

Možemo sa zadovoljstvom da vam saopštimo da je bar sa dva računara razbijena.

Jedan od naših poznatih profesionalnih softveraša koji je želeo da ostane anoniman na PC-ju je postigao 32 000 poena i posle toga brojač je počeo da odbrojava naniže.

Na C-64 organizator emisije „Čip i sedam jarića" **Čedomir Radović** dostigao je 256 linija i posle toga se brojač linija postavio na nulu.

Za razliku od PC-ja, brojač je normalno reagovao kada je bila predana brojka od 32.000 poena.

I još nešto.

Od **Nenada Dunjića**, konstruktora „Ilme 011", dobili smo informaciju da će se „Tetris" uskoro pojaviti i u verziji za ovaj računar.

Preostaje pitanje — da li će i ovaj program jednog dana na gornji način biti razbijen?

Mirisi iz kuhinje

Sve je prilika da ćemo još jednom moći da budemo Dejli Tompson. Ako sve bude kako valja, s početkom olimpijskih igara pokloniće se nova sportska simulacija. Naravno, simulira se desetoboj, disciplina u kojoj D. Tompson briljira i u kojoj će nastojati da na seulskoj olimpijadi uhvati tri zlatne medalje. Njima će i jedna biti dosta, ukoliko pre toga ne iščašimo džojstik.

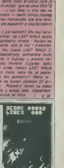
Za septembar se sprema još malo sporta: simulacija fudbala. Videćemo je pod zaštitnim znakom firme koja bude ponudila najviše love. Sensible Software-u. Videćemo ali kako će nam biti? Znajući kako su dosad izgledale igre u kojima se simulirao fudbal, možemo dopustiti sebi i nešto malo skepse.

Van sporta, nostalgija hara ove žeće. Stare igre odevaju se u novo ruho i očigledno prodaju kao pivo. Spremaju se nova izdanja, govori se da će među njima biti FROGGER, GALAXIAN, DEFENDER i MISSILE COMMAND. Sjećate li se Doli Bel?

Nostalgija u daljnje obzorje i jedna sasvim nova igra: THE EMPIRE STRIKES BACK, iliti IMPERIJA UZVRAĆA UDARAC. Film na koji ona nastoji da liči stariji je valjda i od PACMAN-a. Luk Skajvoker i Han Solo imaće posla sa gomilama i gomilama kosmičkih zlikovaca u linearnoj tredimenzionalnoj (i verovatno beskrajnoj) igri, proizvodu iste one ekipe koja je uradila STAR WARS.

Skejtbord kao izvor inspiracije polako presušuje (da kucnem u drvo!) — ali rekla bi se da će ga zameniti nešto slične koturaljke. Isto to, samo malo drukčije (ako je uopšte drukčije). Igra se zove SKATE CRAZY i u njoj ćete — ako vas ne bude mrzelo — vaditi klinca zvanog Fred kroz manje-više iste one terene koje ste zapamtili iz skejtbordskih dana.

Ima li još naivnih? Oni koji veruju, na primer, da LAST NINJA zaista znači poslednji nindža! Nadam se da ih nema, ako je koji i preostao, srušiće mu iluzije LAST NINJA 2. Neme poslednjeg softverske kuće presipaće iz šupljeg u prazno dok budu likvidne. (Uzgred, kako vam se sviđa naslov LAST NINJA 2? Kako može neko ko je poslednji ne biti poslednji? Mene to podseća na čuveni početak SF pripovetke: „Poslednji čovek na svetu sedeo je u svojoj sobi. Odjednom neko zakuca na vrata ..."). V. St.



634

Nekad bilo

Oni stariji sigurno se sećaju igre „Sabre Wolf".

Zbog onih mlađih koji se tek uče računarskoj azbuci ili se po prvi put suočavaju sa iskušenjima uzijanog džojstika započinjemo sa rubrikom koju smo svojevremeno najavili i u kojoj isključivo računamo na vašu saradnju.

*Čast da je otvori pripada **Gordanu Žiži** iz Maglaja, a što se tiče već pomenute igre u svoje vreme bila je veliki hit.*

*Rame uz rame tu je i **Boris Zrilić** iz Banja Luke sa prikazom još jedne stare dobre igre „Jack and Beanstalk".*

JACK AND BEANSTALK

Ako vam je još u sjećanju bajka o čarobnom grahu, ovaj program vas neće razočarati.

Stvar počinje tako što se čarobni grah zelene boje uzdiže do neba očekujući Džeka (odnosno vas) da se uz stabljiku popnete do vrha ekrana. Prvo uzmite sjekiru, potom izbegnite pauka a pri vrhu na vas vrebaju čarobno animirane ptice. Posle toga, opet se penjete do vrha ekrana, ali sada uz kamene biskove. Pri vrhu uzmite vreću s novcem i onda ulevo. Zatim se penjete po zidu od opeka, skačete u kavez, uzimate ptice i polako se spuštate konopcem u četvrti dio. Tamo vas čeka iznenađenje — ogromni uspavani div. Imate malo vremena da zgrabite predmet sa stola i to prije nego što se div probudi, jer u suprotnom sve počinje ispočetka. I onda slijedi rasplet — morate sjekirom da posiječete grah u podnožju pre nego što div siđe. Potom se onoj koji su uspešno okončali misiju slijedi i prijatno iznenađenje u vidu poslednjeg zadatka.

Ali, to neka ostane mala tajna.

Boris Zrilić

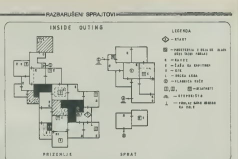
SABRE WOLF

Kralj Afrike je izgubio svoj vučji medaljon, a kako vi želite njegovu kćerku za ženu, on vam naređuje da ga nađete inače ništa od vjenčanja. Pri polasku kralj vam je napomenuo da mu se medaljon pri zadnjem lovu razbio na četiri dijela, što uveliko otežava vašu potragu.

Vi ste u ulozi istraživača koji se probija kroz lavirint sastavljen od biljaka iz džungle. Izgleda kao da su u igru unijeli cijeli botanički vrt. Vaše jedino oružje je mač. Njega ne šaljete, ne bacate, već se normalno mačujete, ubijajući životinje koje vas napadaju. Neprijatelja ima dvije vrste: smrtni i besmrtni. Smrtni su majmuni, ptice, ptice sa krilom šišmiša, miševi, pauci, ježevi i još mnogi drugi, a besmrtni su medvjedi, nosorozi, divlje svinje i kraljevi ljudoždersktih plemena.

U igri postoje i neka pomagala kao što su mali šipovi koji daju dodatne živote, ili orhideje koje se rascvjetaju i žive samo tri sekunde i ponovo nestaju kao da ih nije bilo. Efekti orhideja su sljedeći: plava daje neuništivost i veliku brzinu, bijela vas vraća na normalno, a ljubičasta daje neuništivost, brzinu. Komande su obrnute. Od žute dobijate granicu ste džungla i za to vrijeme nestaju svi neprijatelji sem onih besmrtnih. Crvena daje samo neuništivost. U ovoj igri crtanje mape je neophodno. Mali savjet: ako hoćete da budete u tisi najboljih, morate imati makar jedan dio medalje i sakupljati sve predmete na koje naiđete, a pogotovo vreće sa novcem, dijamante i dijelove istraživačke opreme.

Goran Žiža



INSIDE OUTING C-64

Treba da pronađete dvanaest dijamanata sakrivenih u jednoj kući. Naravno u njoj ima svega i svačega. Pri tom se praktično svaki predmet može vući i gurati (f7), a manji predmeti kao što su čaše, boce ili dijamanti mogu se uzeti (f1), preneti u drugu sobu i tamo ispustiti (f3). Energiju (nema je nikad dovoljno) gubite u dodiru sa pticama, miševima, vatrom u kaminu i bodljikavim biljkama. Od ptica je jedina odbrana kavez — kada ga donesete u prostoriju gde ova živinče leti i ispustite ona će bežati i omogućavati vam da uštedite bar malo energije. Miše treba navesti da pojede parče sira: ispustite parče sira u pravom trenutku ispred njega i bićete mirni dok se eventualno ne vratite u istu prostoriju. Za bodljikave biljke nema leka. Odlikuju se tamno-bison bojom (ostale biljke nisu opasne). Ako su vam zakrčile prolaz, pažljivo priđite i odvucite ih ili postavite neki drugi predmet između vas i opasne saksije i gurajte.

Što se tiče kombinovanja i prenošenja nameštaja iz prostorije u prostoriju nikakvo pravilo ne postoji osim toga da se u sobi sa otvorom na plafonu ne može nalaziti više od pet predmeta.

Kada pronađete bilo koji od dijamanata morate posetiti sobu sa vlasnikom kuće i predati joj ga. Broj preostalih se tada ispisuje na stolu u toj sobi. Energiju obnavljate uzimanjem čaše sa napitkom. Međutim, ako je ispijete u času kada imate dovoljno energije, postižete suprotan efekat.

A sada — gde i kako pronaći dijamante?

Dijamant broj jedan nosi ptica koja za razliku od ostalih leti neuobičajeno visoko. Neophodno je da je vešto manevrišući kavezom naterate da se zaglavi iznad police na zidu. Uz pomoć predmeta koji se nalaze u sobi, a i vaze koja se nalazi u susednoj sobi, lako se sada popeti do nje.

Dijamant broj dva nalazi se ispod ptice zatvorene u desnom štednjaku sa pomičnim poklopcem koji naravno treba izvući.

Dijamant broj tri naći ćete u sefu koji krije slika sa likom vlasnice. Pomerite je levo ili desno i opet ste uspeli.

U prostoriju gde se krije **dijamant broj četiri** ulazi se kroz kamin. Najpre kockom leda ugasite vatru, a zatim izvucite ložišta. Dijamant se nalazi između ormana i zida — da bi do njega doprli potrebno ga je izgurati sa tri ugla predmeta (čaše ili svećnjaci).

Do **dijamanta broj pet** doći ćete u sobi sa otvorom u sredini. Ali u tu rupu treba najpre ubaciti nekoliko predmeta da biste ponovo mogli da se popnete nazad i nastavite igru.

Doći do **dijamanta broj šest** je prilično komplikovano. U odgovarajućoj sobi police sa knjigama zaklanjaju od vas dva stola, miša i dijamant. Prvo onesposobite miša i odvucite ga što dalje, popnite se na policu i ubacite neki predmet između nje i stola. Onda uzmite drugi predmet pa i vi skočite. Sada odvucite drugi sto, preskočite ga, uzmite dijamant i pomoću drugog predmeta nazad. Dakle važno je da se stolovi ne mogu preskočiti sa poda.

Sedmi dijamant nalazi se u kaminu, ali sada to više i nije prolaz.

Osmi dijamant nalazi se u dvostrukom dnu stola za bilijar. Prvo morate da u svaku rupu na stolu ubacite po jednu lopticu. Potom drugim predmetima izgurajte dijamant do ivice stola.

Deveti dijamant ćete naći slično kao i drugi, ali ovde više nema ptica.

Da biste došli do **desetog dijamanta** u pravoj sobi postavite tri stolice paralelno sa stolovima pa onda pomerajući te iste stolove oslobodite prolaz. Dijamant se nalazi ispod krajnjeg levog stola.

Sa **jedanaestim dijamantom** je identičan postupak kao sa petim.

Poslednji **dvanaesti dijamant** nalazi se između dva ormana. Primenite postupak kao kod četvrtog, ali sada vam je potrebno sedam predmeta i posao ste završili.

Na mapi su označeni samo najvažniji predmeti. Tajni prolaz u prostoriju sa ledom krije levi orman i treba ga odvući u stranu. Leda takođe ima i u jednom štednjaku.

I to je sve.

Predrag Stanisavljević

Poukova mreža

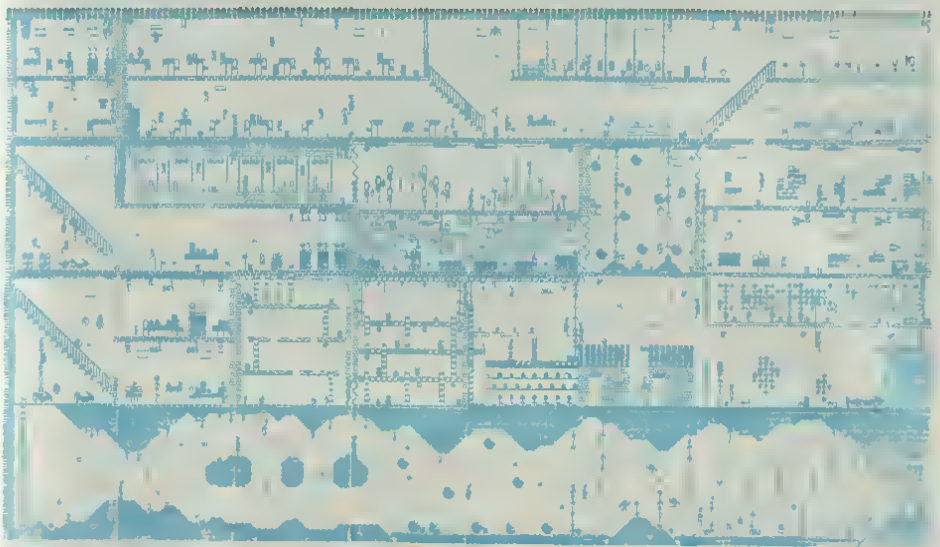
Spektrum — *Tomislav Klemenčić*

GUN SMOKE — DESPERADO

POKE 65119,129 bezbroj života

Komodor — *Bojan Vujošević*

I BALL 2	POKE 40528 POKE 41627 173 POKE 49024,0
RASTAN	POKE 5813 173 POKE 5814 173 POKE 5807,44
DRAK	POKE 3204 255 POKE 2205,255
TRAZ	POKE 54027 173 POKE 54030 173
QUEDEX	POKE 23015 188 POKE 23017 188 POKE 28080,44
MASTERS OF THE UNIVERSE	POKE 36280 44 POKE 36282,44
STAR WAR'S	POKE 47132 173 POKE 47133 173
DROID D	POKE 41305 173
STAR BALLS	POKE 9128 44 POKE 9128,0 POKE 3125 44
ROAD WARS	POKE 4713 173. POKE 4714 173 POKE 5046 66
XORNOR	POKE 15304 177 POKE 15306 177



OUT RUN

Igra je izvanredna vozite se u brzom kabrioletu zajedno sa lepom plavušom pokušavajući da što pre provezete pet staza. Vozite raznim putevima od uskih i krivudavih planinskih do širokih ravničarskih i krivudate između ostalih učesnika u saobraćaju koji sude po vidiomom u igri baš i na mara mnogo za vašu bezbednost. Meni je klasičan promena gustina saobraćaja biranje komandi i start igre (Out Run). Postoje četiri gustina saobraćaja e ja vam preporučujem da odaberete prvu (EASY) koja je najlakša. Od komandi možete birati tastaturu (G-gas A-kočenja O-levo P-desno i M-menjač) a ako koristite džojstik imate na raspolaganju kampeton „interi i kurser

Igra ima 15 staza koje se učitavaju zasebno Prva se učitava samo jednom na početku i stalno ostaje u memoriji Na kraju svake staze birate sledeću iako ste skrenete levo ili desno na rečvanju Raspored staze je sledeći

11 12 13 14 15
7 8 9 10
4 5 6
2 3
1

Na petnaest staza

U mojoj verziji programa postoji beg na 15 staza tako da ju je nemoguće preći Ne znam da li je to krivica pirata ili originalni program ima takva greške Da biste unapred znali šta vas na kojoj stazi čeka dajem kratak opis svake od njih.

1: Okolina je zelena put je se četiri traka. Oko puta su palme i žbunje a uz vas idu folksvageni tube Moj rekord 0'36 4

2: Boja je žuta put uzak i krivudav bez linije uz vas su kamioni i „ferariji Moj rekord 0'32 7

3 Zelena boja pored puta promiču vetrenjače Put je brz se iri traka prepun perišea 911 Rekord 0 41 9

4 Zelena pozadina pored puta promiču kamena kuia Put je delimično nadsvođen što usporava igru Pratićete „ferarije Put se iri traka Rekord je 0'57 1

5 Pozadina je žuta pored puta su drveredi Put ima iri traka vozite između kamiona. Rekord je 1 07 9

6 Okolina je zelena pored puta je drveće Oko vas su bube i „ševroleti Četiri traka Rekord je 1 19 4

7 Boja je ljubičasta put uzan i neoznačen krivudav jer prelazite kroz planinske predele prepoznatljive po panjevima i drvedu pored puta Ometaju vas foticike Rekord je 1 02 2

8 Crvena pozadina stubovi i kuće sa strane perišei Put je širok i krivudav Rekord je 1 04 0

9 Zelena pozadina ometaju vas „foticike pored puta su kameni stubovi ima iri traka Rekord je 1 17 9

10 Pozadina je žuta jure vas perišei pored puta su stubovi za plakate Četiri traka Prilično teška staza Rekord je 0'57 9

11 Okolina je zelena ometaju vam „ševroleti pored puta su table i drveće Širine varira od dva do šest traka Rekord je 1 00 9

12 Crvena okolina okolo su ferariji pored puta kamene kuie i stena koja zalaze u put Put je uzak i bez linije Na putu se ponegde nalaze stubovi Rekord je 0 57 4

13 Bela pozadina put je naizmenično uzak i širok kameni stubovi na putu i pored Lako se izleće a put se povremeno rašva Okolo su ferarij Rekord je 1 02 2

14 Pozadina je zelena okolo su ferarij i perišei Put je širok i rašva se pored puta je gomije i drveće Rekord 0 56 0.

15 Pošto ova staza kod mene ima beg mogu vam samo reći da je pozadina zelena

Ka može bolje

Maksimalna brzina kojom se vaš automobil može kretati je 264 km/h a nju postižete tek posle dužeg neprekidnog držanja pede Pri sudaru se drugim vozilima ili okolnom autu će početi da se sbrde oko sebe sve dok mu brzina ne padne na nulu Trenutak kasnije premestiće se na bližu ivicu puta i opet ćete moći da upravljate njime Ukoliko sudar nije direktan auto će se okrenuti oko sebe a zalim nastavili da ida dalje kao da ništa nije ni bilo Zanimljivo je i to da je proator između dva vozila na susednim trakama dovoljno velik da prodete između njih što trebe koristiti na uskim ravnim delovima

Kada prodete i poslednju stazu (STAGE 9) na ekranu će zatreperiti veliki natpis CONGRATULATIONS a auto će postati neosetljiv na komande Uskoro će se zaustaviti a poeni će se uvećati se 100000 puta prostete sekunde

Ako budete imali dovoljno poena moći ćete da upišete svoje ime u tabelu rekorda Moj rekord je 33279980 se ukupnim vremenom 4 43 9 za svih pet staza Ke je bolji svake mu čast

Na kraju treba reći da je igre zaista izvrsna urađene i da bi jedina mana mogla biti usporenje igre pri pojavi objekata pored puta. To je naročito izpoljeno na četvrtoj stazi gde je staza natkrivena Međutim poznji i jedna dobra strana ovog usporenja počto se sva sporije odvija veoma je teško poginuti zbog čega vam ovu stazu preporučujem

Sve što vam sada preostaje je da pokušate da srborite moje rekorde To će u nekim slučajevima biti lako a u nekim neće Ako nagde uspete poštajte rekord Računarima da najrevnije rangiranju najuspešnijih Outrunner-a Puno prođenih kilometara bez udesa želi vam

Ranko Tomić

Poukova mreža

Komodor — *Elvis Beganović*

DEVIANTS

POKE 19068 0 POKE 20068 0 POKE 2 167 0 SYS 16384

TRANTOR

POKE 9571 252 SYS 6454 — nerc

CODE NUNTER

POKE 8758 173 SYS 4700 — besmrtnost

MYSTERY OF THE NILE

POKE 4329 173, POKE 8122 173 POKE 1256 99 — besmrtnost
Šifre I — PB68472V, II — OC648471P III — PF578170V

SLIMEY'S MINE

Početni skrin pored naziva igre sadrži i opcije za promenu nivoa (tasteri F1 i F3) — ukupno ih ima 99 i opcija za promenu položaja asteroida (tasteri F5 i F7) — takođe je moguće izabrati jednu od 99 kombinacija.

Potom pritiskom na dugme džojstika ulazite u igru

Vidite tri žapca u posudama koji su spremni za lansiranje u kosmos Kada to učinite, žaba se postavlja na početni asteroid za dati nivo Asteroidi sa otvorom očekuju da udete u njih asteroidi bez otvora sa posle svakog preskoka smanjuju tako da se put za vama zatvara Asteroidi sa strelicom vas odmah transportuju nalevo a asteroid sa uplinkom vas prebacuje u drugi deo asteroidnog polja Sve ovo pratite na skaneru sa desne strane ekrana Najbolje je da na početku ne ulazite u asteroide sa više od četiri prostorije jer od broja pređenih prostorija zavisi da li ćete moći da pokupite sve žabe u poslednjoj — vastilje označenoj prostoriji Ukupno treba pokupiti žabe iz četiri asteroida da biste popunili rupu ispred vašeg žapca preko koje stižete do nelene i prelećete u sledeći nivo

Kada uđete u asteroid imate zadatak da uništite određeni broj balona Oni se pretvaraju u dijamante Treba ih sakupljati da bi se prošlo u sledeću prostoriju

Tu vam opet pomažu tasteri

F1 — TRANSBAR je način da stignete na određeno mesto u određenoj prostoriji

F3 — brza paljba koja vam omogućava da lako likvidirate neprijatelje (potreban vam je paket kuglica koji se svremena na vrema pojavljuje na ekranu)

F7 — prelazak u sledeću prostoriju pod uslovom da je broj pređenih prostorija veći od jedinice

Pored dijamanata možete sakupljati pakete kuglica pomorandžu koja uništava zidove u prostoriji i posudu koja šekira balone — onda ih kupite i time obnavljate sebi energiju

U donjem delu ekrana nalazi se broj osvojenih poena broj sakupljenih dijamanata broj pređenih nivoa a i količina paketa kuglica koji vam je na raspolaganju

Posuda na ekranu pokazuje vašu energiju — u svakoj prostoriji možete najviše tri puta ostati bez energije a onda na scenu stupa sledeći žabac.

I to je sve

Dejan Marković

HOT POP

Upravljate simpatičnim gladnim prasetom koje mora da pokupi hranu kako bi se prehranilo Hrana (banana ananasi, torte i slatkiši) postavljena je na različite visine pa prase mora da ide sa platforme na platformu Ali nije sve tako jednostavno Naše prase mora da beži od neprijateljski nastrojenih pajceka koji su nešto tamnije boje jer će jadno ako dođe u dodir s njima izgubiti život Pod je pun rupa tako da prase ne sme da upadne u njih jer će opet, izgubiti život Treba paziti da ne padne sa veće visine jer će se onesvestiti

Kada pokupi svu klopu iz ćoška na uglu ekrana izviriće glava mama gice koja će ga dozivati (skviik) Tada brzo dođite do nje i prelazite na sledeći ekran

U ovoj igri nema skrolovanja već svaki ekran predstavlje jedan nivo Srećno!

Aleksandar Petrović

bejzik

Pomozite svome tati

STRIP POKER

Iako je ovo relativno star program još uvek je zanimljiv naročito za naše očeve Ovaj prilog je namenjen onima koji su proveli po celu noć pokušavajući da svuku Majndi (tako je ime ženskoj) ali im to nije polazilo za rukom

Verzija za koju je ovo namenjeno ima sledeći raspored

— kratak bejzik
— kratak mašinac (Bytes H)
— SCREEN $
— blok bez zaglavlja dužine 42240

Ako imate drugačiju verziju programa jednostavno je prilagodite ovoj

Pustite da prođu prva tri dela pa zaustavite kasetofon Posle toga učitajte asembler i upišite kratki program

```
        ORG     23296
        LD      HL,LOAD
        LD      DE,16384
        LD      BC 100
        PUSH    DE
        LDIR
        RET
LOAD    LD      SP 22527
        LD      IX,23296
        LD      DE 42240
        LD      A,255
        SCF
        CALL    1366
        LD      HL, (23730)
        DEC     HL
        LD      SP HL
        DEC     HL
        DEC     HL
        LD      (23613) HL
        LD      (IY+49),2
        JP      4777
```

Sada otkucajte RANDOMIZE USR 23296 i pustite kasetofon da se učita treći deo Posle učitavanja će se pojaviti kursor u dnu ekrana. Program ćete moći da izlistate ako je učitavanje bilo ispravno

Imate na raspolaganju sledeće mogućnosti

— USR 43306 prikazuje Majndi u početnom položaju
— USR 43152. skida haljinu
— USR 43028 skida grudnjak
— USR 42859 skida do poslednje krpice (a to nam je bio cilj)

Ako želite program možete snimiti drugačije tj u 2 dela

— deo 1 bejzik koji bi po auto-startu (npr od linije 9999) učitavao sledeći blok i startovao sa RUN

— deo 2 blok snimljen sa CODE 42745 22790

U — bloku pre učitavanja drugog dela treba zadati CLEAR 42744!!! Takođe možete vršiti i druge izmene npr u cilju ubrzanja programa jer mu je najveća mana „tenjost" posebno pri očitavanju tastature

SHANGHAI

Ovo je kompjuterska adaptacija drevne kineske društvene igre stare preko 2 000 godina Nakon startovanja ugledat ćete polje za igru sa opcijama Opcije su REMOVE UNDO FIND MENU CANCEL Igra se uz pomoć strelice koje se nalazi u gornjem lijevom uglu i već navedenih opcija Dovedite strelicu na opciju MENU i pritisnite pucanj Ugledat ćete niz novih opcija

LOAD / SAVE OPCIJA — služi za snimanje pozicije na traku i ponovno učitavanje igre sa traka
CHALLENGE OPCIJA — igra za dva igrača
TOURNAMENT OPCIJA — određivanje vremenskog ograničenja u igri
RESTART OPCIJA — prekidanje igre i započinjanja od početka

Polje za igru se sastoji od osam redova u kojima su naslagane karte Ukupan broj karata je 144 Karte su poslagane tako da tvore piramidu Dakle vide se one karte koje su na vrhu piramide a ispod njih se nalaze druge Vaš je zadatak da sakupljate parove karata Kad sakupite par on nestane sa polja i oslobodi karte koje su bile ispod njih Vi možete da pokupite samo one karte koje su slobodne

Na primjer ne možete uzeti kartu iz sredine reda ako se oko nje nalaze druge karte. Karta mora biti slobodna bar sa jedne strane

Postoji sedam vrsta karata KARTE SA SIMBOLIMA LOPTE, TRSTIKA LJUDI ZMAJEVI STRANE SVIJETA (W E S N) OZNAKE GODIŠNJIH DOBA (VJEVERICA=PROLJEĆE, SUNCE=LJETO LIST=JESEN SNJEŠKO=ZIMA) I CVIJEĆE

Karte označene simbolima lopte trstike i ljudi su obilježene brojevima od I do 9 Njih sakupljate tako što morate pronaći par sa istim brojem (na primjer dvije karte pod brojem 5 sa simbolom trstike)

Karte sa zmajevima i stranama svijeta također se skupljaju stvaranjem istih parova, samo je razlika u tome što te karte nisu obilježene brojevima (na primjer morate pronaći dvije karte sa slovom N — sjever ili dvije karte sa lednim zmajem)

Karte sa simbolima cvijeća i godišnjih doba su rijeđe pa zato nije potrebno pronaći dva ista cvijeta nego tu se par može složiti i sa različitim kartama (npr dva različita godišnja doba=LJETO — ZIMA)

Pažljivo proučite igračko polje i kad pronađete dve karte koje tvore par dovedite strelicu na njih i pritisnite pucanj na džojstiku Obadvije karte će pobijeliti Zatim idite na opciju REMOVE i pritisnite pucanj Ako ste sastavili dobar par, karte će nestati sa polja. Ako ste izabrali dvije karte koje ne mogu tvoriti par (npr ZMAJ i LOPTA) kompjuter će vas odmah obavijestiti da to nije prikladan par rečenicom NOT A MATCH I vi ćete morati poništiti pokušaj pomoću opcije CANCEL i probati pronaći neki drugi par karata

Ako vam to ne pođe za rukom pređite na opciju FIND i kompjuter će vam pokazati nekoliko parova Ipak, ako ste pošten igrač a vjerujem da jeste onda ovu opciju nećete mnogo upotrebljavati

Savjetujem vam da počnete oslobađati srednje redove jer je u njima najviše karata i sve su naslagane jedna na drugu Kad uspijete igru završiti to jest kad uspijete naći parove svim kartama i pri tome osloboditi polje za igranje od karata ukazat će vam se predivna slika koja dočarava ugođaj daleke Kine Nabavite ovu odličnu staru društvenu igru i predahnite od raznih napadača iz svemira

Božidar Alajbegović

A.C.E. II

Igra A.C.E. postigla je zapažen uspjeh pa je firma Cascade izdala i nastavak. A.C.E II je igra koja se više ne bi mogla zvati borbenim SIMULATOROM letenja jer elementi pravog simulatora zapravo u ovoj igri i ne postoje Potrebno je samo stiskati pucanje i okretati džojstik u smjeru u kojem hoćemo da idemo Ova igra nam nudi opcije za igranje jednog ili dvojice igrača Ako igrate protiv prijatelja ekran će biti podijeljen na dva dijela i prvi igrač će koristiti gornju polovinu ekrana a drugi donju Inače mapa se uvijek iscrtava u gornjoj polovini Sama igra nam nudi dva scenarija

1 Borba dva lovca gdje je cilj samo uništiti onog drugog U ovakvoj borbi ste naoružani mitraljezom i raketama na toplotno navođenja kratkog dometa

2. Uništenje protivnikovih položaja na zemlji u čemu vas svesrdno ometa protivnički avion Ovaj je prije polijetanja možete izabrati naoružanje koje se sastoji od

— najviše 8 raketa sa toplotno navođenja kratkog dometa

— najviše 8 raketa sa radarskim navođenjem

— najviše 2 rakete zrak—zemlja (ako uzmete ove ne možete uzeti rakete sa radarskim navođenjem)

Nakon početnog skrina slijedi otimen meni u kojem možete mijenjati parametre igre scenario detekcije pada na malim visinama nivo igre kompjutera broj života koliko puta morate pogoditi avion (protivnički naravno) da bi ga srušili

Parametre mijenjamo pritiskom na pucanje Najbolje je da uzmete da ga tri puta morate pogoditi da bi eksplodirao jer je kompjuter prilično vješt pilot

Nakon svega izabranog u verziji igre za komodor plus/4 slijedi šifra koju nisam dobio od pirata a nalazi se upakovana u originalnoj kazeti Ali imam sreće pa skoro uvijek pogodim (možete pogriješiti čak tri puta) samo jedan put me sreće napustila Zatim se pojavljuje skin sa lijepo nacrtanim avionima i kompjuter očekuje da pritisnete pucanje za polijetanje

U toku leta u svakom trenutku možete da uključite mapu pritiskom na X. Oružje mijenjate tipkom E. Nišan mitraljeza uvijek se nalazi ispred vas a nišani raketa pojavljuju se samo kad se protivnikov avion nalazi ispred vas Ako protivnik ispali na vas projektil kompjuter će vas upozoriti (i zvučno i porukom) a kretanje projektila vidjet ćete na radaru Kada vam projektil bude za petama, stisnite Q da biste omeli njegovu putanju Tako činite sve dok ga ne budete više vidjeli na radaru Inače uz radar od instrumenata na ploči još se nalaze kompas pokazivači snage brzine visine položaja aviona i prazan prostor ispod radara za poruke Sa strane ekrana nalaze se bodovi i oznake pilota (player-computer) Ako ste dobro obavili posao očekuje vas poruka mission completed povratak u bazu i upis imena na listu petorice najboljih asova neba

A.C.E. II pruža osta i ostalo zabave ako igraju dvojica igrača jeden protiv drugoga dok je u igri za jednog igrača ipak bolji A.C.E ! *Marijan Franović*

637

THE RACE AGAINST TIME FOR SPORT AID '88

Sigurno ste čuli za akciju Sport aid 88 koja će započeti 11 septembra istovremeno na svim kontinentima i kom pretv vremena a poslužiće kao pomoć ugroženoj djeci Afrike. Baklja prijateljstva krenuće iz Južne Afrike i svijetom će je pronijeti poznati sportaši a u akciju su uključeni i svi stanovnici svijeta. Majstori programiranja Oliver Twins postavio te ja u ulogu Omare Khalile — sjajnog afričkog trkača. Zadatak mu je da se bakljom prijateljstva obiđe sve kontinente i na kraju upali olimpijske plamenove ili podigne olimpijske zastave. Olimpijske plamenove upaliti će u Aziji Australiji Evropi i Sjevernoj Americi, a olimpijske zastave podići će u Africi i Južnoj Americi.

Tipke u verziji za spektrum su slijedeće Z-lijevo X-desno K-skok M-teleportiranje SPACE uzimanje/ostavljanje predmeta. Sa početne pozicije kreni dva ekrana lijevo i ugledaćeš izlaz na seredrom. Pokazaće ti se karta svijeta i uz pomoć aviona odaberi na koji kontinent ćeš putovati. Kada to učiniš krećeš u potragu za olimpijskim bakljama gdje ćeš upaliti plamenove ili za kopljima na koje ćeš podići zastave. Na raspolaganju imaš pet minuta a za svaki pronađeni pješčani sat dobivaš dve dodatne minute vremena.

Obilazeći kontinente imaš priliku vidjeti najkarakterističnije gradove i države i značajna mjesta Pariz London i Berlin u Evropi Indiju Kinu SSSR i Japan u Aziji New York, Hollywood i Toronto u Sjevernoj Americi Brazil i Argentinu u Južnoj Americi Melbourne i Adelaide u Australiji Egipat u Africi

Na svakom kontinentu kupiš i određene predmete koji ti mogu pomoći u određenim trenucima. Odjednom možeš nositi samo jedan predmet. Evo primjera: otputuj u Aziju i kreni lijevo nekoliko ekrana. Doći ćeš do zida i vidjeti da se na njemu nalazi pješčani sat. Ako pokušaš skočiti na zid ustanovit ćeš da je to nemoguće. Vjerovatno ti je potreban neki predmet pomoću kojeg ćeš to ostvariti. Zato kreni u potragu po ostalim kontinentima. U Sjevernoj Americi uđi u prvi teleport i kreni lijevo. Prođi kroz Hollywood i na kraju ćeš ugledati stolicu (STOOL). Možda je baš ona rješenja? Vrati se u Aziju na isto mjesto i postavi ja pokraj zida. Kompjuter te automatski diže gore. Pokupi pješčani sat i kreni lijevo. Ugledaćeš olimpijsku baklju. Priđi joj i malo sačekaj. Vatra se pali — obavio si šestinu zadatka. Napomena: teleporti su označeni svjetlećim strelicama.

U Africi Australiji i Evropi možeš do određišta doći bez ikakvih predmeta dok će u Sjevernoj i Južnoj Americi to ići malo teže (nemaju ti prometni znak odnosno voda). Pokušaj sam pronaći predmete koji će ti otvoriti put. Većina potrebnih predmeta nalazi se baš na kontinentima gdje bez ikakvih teškoća dolaziš do cilja, ali su predmeti vrlo dobro zakamuflirani i zaštićeni određenim klopkama. Primjer: doputovaš u Aziju, kročiš i u Sibir i onda vidiš da je tvoj junak pao kao pokošen. Pitaš se zašto? E sad ja tebe pitam znaš li kolika je tamo temperatura? Duboko ispod nule. Zbog toga ti treba kaput (COAT) a naći ćeš ga u Africi. Sada slobodno možeš proći dalje i možda pronaći neki novi predmet. Ako prolaziš kroz Berlin trebaš kišobran (UMBRELLA) koji ćeš naći u Australiji. Kišobran koristi i na mjestima gdje kaplje voda.

Ako trebaš redom sačepiti uzmi čep (PLUG). On se nalazi u Evropi. Došao si do nekog žbuna ili drveta i vidiš da postoji put koji vodi dalje ali ti ne možeš proći. Upotrebi sjekiru (AXE). Naći ćeš je takođe u Evropi. Možeš pokupiti i francuski ključ, vreću sa pijeskom (SAND BAG) itd. Ako ti sve otkrijem igra će izgubiti na draži (a to bi bila šteta). Savjetujem ti da najprije obiđeš sve kontinente, ustanoviš eventualnu vezu među predmetima i klopkama pa tek onda da kreneš u završni pohod. Pazi još i na vrijeme i 100%-ni uspjeh ja garantovano tvoj.

Zlatan Hamzić

TOBRUK 1942

C64—128

Evo još jedne od malobrojnih simulacija značajnih bitaka II svetskog rata rađenih za komodor 64. U scenariju ove vrlo dobre strateške igre dosta je verno prikazano stanje na bojištu severnoafričke Afrike u to vreme.

U toku igre na ekranu se nalazi mapa jednog dela Libije koji se inače zove Kirenajka i gradovi Gazala Acroma Knightsbridge El Adem Bir Hacheim i Tobruk koji je najvažnije strateška tačka ove bitke. Narevno borbu vode sile osovine (Axis) i saveznička vojska (Allied)

Sile osovina predstavljaju nemačko-italijanske trupe u čijem sastavu se nalaze jedna laka i dve oklopne nemačka divizija, tri motorizovane (Ariete Trieste Littorio) i četiri pešadijske (Pavia Brescia Trento i Sabratha) italijanske divizije. Ovim jedinicama narevno komanduje nalzbežni Romel.

Savezničku vojsku sačinjavaju jedna indijska motorizovana brigada tri južnoafrička brigade i jedna divizija, i britanska vojska koja u svom sastavu ima pet oklopnih dve gardijske jednu motorizovanu i tri pešadijske brigade.

Zadatak Romelovih trupe je da zauzmu Tobruk vrlo važnu britansku bazu koji je okružen sa oko 170 bunkera i time odseku britanske kopnene od pomorskih snaga dok je zadatak saveznika da sačuvaju zauzete gradove i Tobruk, koji je od velikog strateškog značaja za dalji razvoj borbi u tom delu Afrike.

Izvor snabdevanja Romelove vojske su tzv SUPPLY DUMP jedinice koje se mogu kretati i one obezbeđuju snabdevanje u razmaku do pet polja dok se saveznička vojska snabdeva (pod istim uslovima) iz gradova koji su u početku svi pod njenom kontrolom. Znači izvor snabdevanja protivničkih jedinica je važan strateški cilj.

Na početku igre između protivničkih strana postavljena je linija minskog polja koja otežava Romelovim jedinicama da izvrše napad na gradove. Jedinice gradovi i štabovi vojski su predstavljeni kvadratima na mapi i nije ih teško razlikovati.

U donjem gornjem uglu ekrana se nalazi tzv prozor koji prikazuje stanje svake jedinice na koju postavite kursor i sadrži sledeće podatke

INF — pešadija
ART — artiljerija
AFV — oklopne snage
SUP — podrška
MOB — mogućnost kretanja

Neke od jedinica nemaju gore navedene formacije zbog prirode njihove vrste (pešadijske divizija nema oklopne snage itd.)

Igra se odvija kroz četiri osnovne faze

MOVEMENT PHASE

U ovoj fazi pomerate svoje jedinice do željenog mesta u zavisnosti od njihove sposobnosti kretanja (MOB) pri čemu je važno znati da svaki dodir sa objektom koji nije vaša jedinica zaustavlja kretanja

ASSAULT PHASE

Ovo je faza u kojoj možete napasti neprijateljske jedinice ili gradove sa kojima imate dodirnih tačaka uz napomenu da sa više jedinica možete napasti jednu ili sa jednom više protivničkih jedinica

RESUPPLY PHASE

U ovoj fazi vršite snabdevanje vaših jedinica ranije navedenim formacijama (INF AFV SUP). Snabdevanje se ne može izvršiti ako je jedinica suviše udaljena od izvora snabdevanja

COMMAND PHASE

U ovoj fazi vršite raspoređivanje snaga u vazdušnoj (Air Support) i podršci inžinjerija (Engineer Support) čiji su elementi predstavljeni sledećim crtežima

— crtež tri aviona predstavlja superiornost u vazdušnim okršajima

— crtež aviona koji ispušta bombu je opcija pomoću koja možete bombardovati bilo koje mesto na mapi (Select Air Strike) ako joj dodelite dovoljan broj poena (u početku 10).

— crtež aviona koji bombarduje prugu predstavlja opciju u kojoj otežavate protivniku da vrši snabdevanja tako da dobije mnogo manje zaliha nego što mu je potrebno

— crteži oklopnog vozila sa antenom i vozila koje ispušta dim predstavlja jedinice koje pomažu u kretanju maskiranju i vezi osnovnih jedinica i ova opcija nije od preterano velike važnosti

— crtež mrtvačke glave je vezan za postavljanje ili raščišćavanje minskog polja (Select Target Minefield) i zahteva veći broj dodeljenih poena raščišćavanje minskog polja mogu vršiti sile osovine a njegovo postavljenje savezničke snage

Boje crteža su vrlo značajne. Crvena boja označava da je u toj oblasti neprijatelj snažniji, žuta boja označava da je uspostavljena ravnoteža a zelena boja znači da ste superiorniji u takvom vidu podrška

Ovu igru možete igrati i protiv kompjutera koji u tom slučaju preuzima kontrolu nad savezničkim trupama a vi ste u ulozi Romela

Istorijska je činjenica da su sile osovine sa Romelom na čelu 1942. u ovoj bici osvojile Tobruk zarobile veliki broj savezničkih vojnika i zaplenila desete ratnog materijala ali u ovoj kao i u svim sličnim igrama mogući su svakakvi preokreti i iznenađenja

Zoran Milivojević

Poukova mreža

Amstrad CPC *Jasmin Halilović*

ARMY MOVES 1

```
10 rem compacted by Futuresoft
20 memory &331a load "army1
30 poke &29c0,0 call &331b
```

ARMY MOVES 2

```
10 rem compacted by Futuresoft
20 rem clave de acceso=15372
30 memory &227a load "army2
40 poke &2646,0 call &227f
```

GREAT ESCAPE

```
10 rem compacted by Horlik
20 openout "c" memory &23f load "great escape" &240
30 poke &266 &9e poke &2a7 &c9 call &258
40 poke &785f &ed poke &7980 &8e poke &7982, &c9
50 mode 1 call &81al
```

ROCKRAID

```
10 rem compacted by Futuresoft
20 memory &3004 load "rockraid
30 poke &5c63,0 call &3005
```

THE MYSTERY OF THE NILE

```
10 rem compacted by Satansoft
20 openout "c" memory &2ff load "mystery
30 poke &784,0 poke &26db,0 poke &2910,0
40 poke &4058,0 poke &4246,0 call &8did
```

Komodor *Elvis Beganović*

BAZOOKA BILL

POKE 27727,173. POKE 27728,234 — besmrtnost

THE VIKINGS

POKE 2180,0 POKE 29434,173. SYS 2084

SUPER STUNT MAN

Još jedna igra u kojoj je slavna uloga poverena moćnom vozilu Džemsa Bonda. Vaš zadatak je da provedete vozilo kroz jedan nivo pazeći pri tome na dve osnovne stvari

• Vreme — Što igra više odmiče procena štete na vozilu je sve veća i na kraju se sve završava jednim životom manje

• Životi — imate ih tri i nastojte da se što duže održite te nije tako lako kad što se čini na prvi pogled i biće prava umetnost da završite igru bez poukova

Svaki nivo ima svoje karakteristike

1 THE DESERT SCENE — Nalazite se u pustinji prepunoj kamenja i svakojakih opasnosti Treba da vozite i brzo i pažljivo Pazite da se ne nađete u ćorsokaku

2. SPEED BOAT CHASE — Očito da je vaše vozilo jedne vrste amfibije U ovom nivou upravljate gliserom pazeći pri tom da ne udarite u neko ostrvo ili da na naletite na neprijateljske bombe koje izbacuju ostali čamci

3. THE FOREST SCENE — Iz vode pravo u šumu Ovde imate zadatak da se izvučete sa što manjom štetom Broj kamenja i grmlja je povećan a tu su još i opasne vikendice(?) koje vam blokiraju put One vam ujedno i pomažu

4 CANYON JUMP — Ovaj nivo predstavlja jednu vrstu damove između dva teža nivoa Nalazite se u kanjonu punom akaksonica Pazite startno vreme je smanjeno Ipak ne bi trebalo da imate probleme ukoliko koristite TURBO pogon koji se dobija prilikom na taster za pucanje

5 CITY CAR CHASE — Vozite se u gradu punom automobila i zgrada Zgrada su postavljene na ulici a neprijateljski automobili su svuda oko vas Pojavljuju se i mostovi koje treba koristiti da ne biste izgubili život.

6. WILD WATER SCENE — Pred vama je novi izazov Treba se što više uspeha preći divlje vode pazeći pri tom na mnoga stešnjanja neprijateljske bombe i ostale prepreke Neprijatelji nastoje da vam oduzmu i poslednji život i time vam onemoguće da vidite VELIKO FINALE.

7 GRAND FINALE — Velike finale je pred vama. U ovom nivou skupljene su sve najteže discipline iz prethodnih nivoa Tu su ponovo zgrade mostovi vikendice i neprijateljska vozila i ovde je najbolje da koristite mostove kao u petom nivou.

Igru nije lako preći bez poukova Zato prilažem program koji treba otkucati umesto kojzika Sa ovim programom imaćete bezbroj života a ako vam i to nije dovoljno liniji 3 dodajte POKE 36532 183.

```
1 CLEAR 24999: LOAD "" CODE: LOAD
  "" CODE
2 POKE 25517 0
3 RANDOMIZE USR 25000
```

Ivan Mirčevski

PANIC EXPRESS

Atari

Pljačkaši su opljačkali voz, ubili mašinovođu i u lokomotivi postavili tempiranu bombu koja će eksplodirati u roku od dva minuta Kao svaki dobar železničar vi krećete u opasnosnu misiju. Igru počinjete sa poslednjeg četrnaestog vagona i imate cilj da stignete do kraja lokomotive gde je smeštena bomba

Krovovi vagona po kojima se krećete su prepuni raznih manje ili više smrtonosnih smicalica Najteže vagoni

Vagon broj 10 — Ostavite na mjestu gde vas je kompjuter ostavio sačekajte da balon sa kukom prvi put prođe a zatim (kad drugi put naiđe) skočite (džojstik od sebe) i balon će vas preneti preko rupe

Vagon broj 9 — Da biste ga prešli pomjerite se malo (ne previše!!!) desno zatim lijevo i skočite Sačekajte pogodan trenutak i krenite Bez prestanka idite do pretposlednja prepreke Malo sačekajte pa je pređite

Vagon broj 7 — Prve dve prepreke ćete lako preći no sa trećom ćete biti malih problema. Potrebno je ići iza te prepreke uporedo sa njom na malom rastojanju Kada ona bude išla prema desnom kraju vagone naglo se okrenite i skočite Stvar je rešena Za ovaj vagon treba dosta vježbe

Vagon broj 5 — Sačekajte da vam se dim približi skočite okrenite se i kad vam se dim opet približi skočite Sad sačekajte balon i uhvatite se za njega

Vagon broj 4 — Skočite i priđite do prve ograde Tu sačekajte da helikopteri prođu skočite i glavno ste prešli

Vagon broj 2 — Kao i 9 a tim što morate povremeno da čučnete (džojstik sebi)

Vagon broj 1 — Kad prvi put naiđu oblačići pomjerite džojstik sebi pa odmah od sebe Zatim malo priđite levom kraju Drugi put se sagnite i kad ustanete sačekajte 2—3 sekunde pa skočite

Vagon broj 00 — Lokomotiva Oblačići dima dolaze u parovima Kad vidite da vam se približava prvi oblak prvog para skočite mnogo ranije no obično da biste imali dovoljno vremena da izvedete i skok za drugi oblak Teško je i sa drugim parom Dodirnite odžak lokomotive i time ste demontirali bombu

Još nekoliko savjeta

— ŽURITE —

— Ako vidite da ste upali u nezgodnu situaciju bolje izgubite jedan od sedam života nego previše vremena

Ivan Bogdanović

TOP DUCK

C-64

Na početku birate nivo od kojeg hoćete da počnete (najviše od devetog nivoa) i brzinu kojom će se odvijati igra. Najbolje je početi od prvog nivoa i brzinom 3 jer je to najsporije Meni je dosad poznato 19 nivoa Cilj igre je da u ulozi patka poubijate sve neprijatelje i pređete u sledeći nivo. Što je najzanimljivije nemate nikakvog oružje sem svog kljuna kojim trebate doroniti kamenje na neprijatelje U tom plemenitom poslu vam pomaže specijalno kamenje koje se aktivira kao i druge ali se red tog kamenja odronjava jedan za drugim Lako ćete ga prepoznati jer je na njemu nacrtan grom U poslu vam na neki način pomaže i neprijatelj Kada odroni te neki kamen koji mu onemogućuje da dođe do vas i doručkuje an da taj kamen obnoviti

U svakom nivou vas proganja šest neprijatelja iste vrste a jedan od njih ide istom brzinom kao vi dok ostali idu sporije

Neprijatelji su raznovrsni kao i u svakoj arkadnoj igri **Pčele** vas proganjaju u 1 8 i 15 nivou **Pingvini** sa južnog pola vas jure u 2 i 12 nivou. U 3 i 11 nivou ubijate **ježeve** **Čistačice** vas jure u 4 10 i 17 nivou **Hobotnice** su izašle na suvo i ganjaju vas u lavirintima 9 13 i 18. nivoa Nedužne **čistače** morate da isprašujete u 9 14 i 19 nivou Iz Sahare su stigle **zmije** i uželele se pačene patke pa vas jure u 7 9 i 18. nivou

Do poslednjeg nivoa možda iskrsnu još neke nakaze Savetujem vam da odmah iskoristite gromovsko kamenje a onda drznike namamite ispod kamenja i — dobićete presovane neprijatelje

Ne ubijajte nedužne životinje jer mogu da vas tuže društvu za zaštitu životinja a onda sta nadrljali bez obzira što ste patka.

Simonović Miroslav

PSYCHO PIG UXB

Radi se o prasićima koji se iz nama nepoznatih razloga ne podnose pa se nemilosrdno tamane bombama Kontrolišeš prase koje se sa svojim drugovima nalazi na polju po kome su razbacane crne vremenske bombe Na teren da istrčati sudija koji se razlikuje po tome što nosi ragbi majicu Kada on zvizne u pištaljku igra kreće i prasići počinju da dacaju bombe On da izaći ponovo da postavi nove bombe. Pazi da ga ne razneseš bombom jer ćeš biti kažnjen

Podle je bacati gasne bombe Kada baciš jednu od njih tvoje prase će staviti gas masku a iz bombe da pošto eksplodira izaći nevidljivi uspavljujući gas koji da deluje i na svako živo prase Pošto si tako lukavo uspavao svoje opasne konkurente mirno možeš jedno po jedno da dižeš u vazduh

Odbijajuća bomba je veoma opasno oružje Kada je baciš, ona će se odbijati da zidove (krajeve ekrana) s najverovatnije će se zaustaviti kod tebe Ako se ne skloniš na vreme Neki prasići su napravljeni do metala i treba da se pogode tri ili četiri puta pre nego što odlete u vazduh. Veoma su nezgodni kada ih na ekranu ima u velikom broju što se često događa na višim nivoima.

Kožno odelo štiti te od udara bombe ali i ono na žalost ima svoj vek trajanje Otporno je samo na jedan pogodak bombom posle toga se gubi ali ti ostaje jedan život sačuvan Bombe počinje da odbrojava do broja koji je zapisan na njoj Kada broj dođe do nula bomba eksplodira Na nižim nivoima startni broj odbrojavanja je 10—75 ali što je viši nivo to je broj na bombi manji Igru je moguće igrati i u dvoje a tim što na kraju pobeđuje onaj kome je ostalo više života Da bi što lakše potamanio prasiće dajemo objašnjenje svih ikona koje možeš uzeti

— **Snažnije bacanje** Pomaže da dalje baciš bombu Ovo nije neka prednost ako je bomba odbijajuća jer će se vratiti (u većini slučajeva) i raznetі te

— **Ubrzivač kretanja** Daje ti dvostruku brzinu kretanja. Ovo je posebno korisno na bonus nivoima koji se pojavljuju između običnih nivoa. Na bonus nivoima ti je cilj da POLJUBIŠ prasiće koji izlaze iz kapada Naravno da je vremenski rok za ljubljenje prasića ograničen Prema tome što ih više poljubiš to će ti bonus poeni biti veći

— **Gas** Uspavljuje sve prasiće na ekranu

— **Mega bomba.** Ako ovu bombu staviš negde na ekran i okružiš je bombama one da ih u jednom momentu sve aktivirati, izazivajući ogromnu eksploziju

— **Kožno odelo** Štiti te do udara bombe (eksplozije)

— **Ranac** U njemu možeš držati više bombi! Pazi samo da u ranac ne stavljaš bombe kojima je fitilj skoro dogorio jer može doći do neželjenih posledica.

— **Poeni.** Daje ti danas 1000 poena

— **Pametna bomba** Ovo je najkorisnija stvar Raščišćava ceo ekran

Ovo bi bile sve opcije ovog izvrsnog programa Želimo vam prijatno uspokojavanje prasića posebno za stolom

Vladimir Janković

FIREFLY

Scenario je manje-više klasičan svemirskim brodom letite kroz lavirinte napucavajući brojne neprijatelja Međutim da je sve u tome igra bi bila ima manetena što ni ukom slučaju nije

Nakon definisanja komandi pojaviće se ekran podeljen na 45 polja Većina ih je prazna dok su neka popunjena sličicama planete i plavim trouglovima Sasvim desno nalazi se sličica poluge obojena zeleno U ovom delu igre upravljate pčelom kojom možete da stupite na sva polja sem onih sa planetama. Prema tome cilj igre je da dođete do poluge pretvarajući pojedine polja bilo prazne ili sa trouglovima u bela kvadrata sa ucrtanim krugom Ako pritisnete fire dok stojite na polju sa dva trougla. prikazaće vam se pet ikone od kojih su dva pozitivne a tri negativne Uz malo veštine i puno sreće pogodićete neku do pozitivnih ikone nakon čega to polje iznajmiće u kvadratno belo U suprotnom se sva polja izmešaju ili zamene mesta i najčešće upropaste sav vaš dotadašnji trud

Prilikom svemogući fire dok ste na praznom polju ulazite u glavni deo igre u koma je od velikog značaja kontrolna tabla u donjem delu ekrana Sasvim levo je potpuno beskoristan displej — više služi za ukras nego što ima neki stvarni značaj Desno su pokazivači količine goriva i energije a u sredini daleko najkorisniji instrument — mapa lavirinta. Uzgred svako prazno polje ima različit lavirint. Pored mape videćete omanji ekrančić čija je funkcija da izveštava o eventualnim dodacima stečenim tokom igre U donjem desnom uglu je brojač grozdova (o njima malo kasnije) Naravno i tu je i neizbežni score

Prateći mapu dolatite svojim svemirskim brodićem do jedne do četiri neprijateljske ratne baze Njih ćete razlikovati na mapi po tome što baze trepću. Uzdajući se u spretnost i lepo izveden štit pokupite četiri grozda koji se s vremena na vreme pojavljuju iz baze. Tada ćete čuti karakterističan zvuk Stanite iznad baze Ekran se briše i ponovo vidite dva ikone koje naizmenično trepću Nagazite fire u trenutku dok leva sličica svetli i — baza je neutralisana. Isto uradite sa preostale tri baze čime ćete oslobodili još jedno polje na putu do cilja Ponekad se baza nalazi ograđene zidom U tom slučaju je korišćenje teleporta neophodno Teleporti se na mapi razlikuju do samih baza time što sporije trepću Po ulasku u teleport naći ćete se okruženi raznobojnim kvadratima. Vaš brod postavljen u centru reška osvetljavajući kvadrat neposredno ispred kljune Na vama je da tri puta zaredom pogodite plavi kvadrat sa sto trade dosta koncentracije i brzine Za kraj da vam kažemo da energiju možete obnoviti kada izlete dva grozda a vi već imate tri u rezervi Prvim otvarate bazu a drugi se pretvara u kapljicu Pokupite je i — bićete kao novi

Igor Ćopić i Predrag Ćulum

PIRATES

O ovoj izuzetno lepoj ali složenoj strateškoj igri već je pisano ali ne i dovoljno pa vam je zato sada detaljno prikazujemo

O čemu se radi?

Pre svega o izvanrednoj rekonstrukciji (izgleda da se i to može u računarskim igrama) jedne uzbudljive istorijske epohe kada su vodama Karipskog mora krstarili odvažni pomorci pustolovi a ponekad i zlikovci i pod cinom gusarskom zastavom vodili nebrojene male ratove ponekad i privatne i to u samim začecima onoga što danas nazivamo kolonizacijom Amerike.

Od šegrta do bitange

Šta je cilj?

Da postanete KING'S ADVISOR kao što je to selsla pošlo se rukom jednom Henri Morganu mada ni Francis Drejk i slična bratija nisu daleko

Idemo od početka

Kada igru učitate (postoji samo disketna verzija) prvi meni vam nudi tri opcije.

Možete početi novu karijeru setim nastaviti snimljenu igru ili komandovati čuvenom ekspedicijom Nova karijera podrazumeva standardnu igru (to je godina 1660) a potom imate mogućnost da u sledećem meniju izaberete još pet istorijskih perioda.

Komandovanje čuvenom ekspedicijom takođe nudi šest različitih igara (radi se o čuvenim bitkama kao što je bila ona kod San Huana, ili pljačka Marakajba) — ali sve u svemu ti scenariji se razlikuju samo po brodovima koji su tada plovili morem Pritom imate različit broj ljudi a kako smo rekli da je igra u suštini strategijska sigurno da na njen tok to ima i odraza

To ostavljamo vama a mi ćemo se pozabaviti standardnom igrom.

Sledeći meni vam nudi da se opredelite da li ste Englez Španac Holandanin ili Francuz.

Nekih velikih razlika tu nema jedino što su vam polazišta — odnosno matične luke na drugom mestu a začin svemu su scenariji u kome su autori igre zaista pustili mašti na volju pa vam se može desiti da ste francuski hugenot koga je prognao strašni kardinal Rišelje ili opet nesrećni engleski mladić čiji je otac, recimo, propio imanje bacio porodicu u dugove i sada vi ispaštate njegove grehe dirinčeći na nekoj plantaži u Karibima i slično

Potom birate težinu igre — možete biti APPRENTICE (šegrt), JORNEYMAN (kalfa) ADVANTURER (nešto nalik majstoru) i SWASHBUCKLER (doslovce, okorela morska bitanga)

Zatim birate veštinu

Od pet mogućih — mačevalac navigator topdžija brodski lekar i šarmer(ovo poslednje ne treba da vas čudi jer se u igri pojavljuju i lepotice u vidu guvernerskih kćeri) savetujemo vam da odaberete mačevanje — najkorisnije je.

Sada ste na putu da okupite svoju gusarsku družinu, ali sledi i kviz koliko poznajete prilike u kraju Postavlja vam se pitanje u kome periodu je španski srebrni karavan ili kraljevska tresor flota u određenom gradu

Dajemo vam kompletnu dokumentaciju Ako odgovorite potvrdno biće vam mnogo lakše da savladate u mačevanju vašeg konkurenta za mesto kapetana broda.

I evo prvog saveta za mačevanje.

Ne može se nešto posebno preporučiti — morate sami mnogo toga da savladate, osnovno kada zamahnete mačem, pustite džojstik. Tada ćete udariti Pritisak na FIRE pojačava snagu udarca.

Ako izgubite taj prvi duel — ugasite računar i počnite iz početka. Karijera koju bi eventualno nastavili baš ništa ne obećava

Ako dobijete — snimite fajl pa će to prvo i opasno mačevanje odsad pa nadalje izbeći

U nekim istorijskim periodima mačevanje na početku nema, a isto važi i za velike ekspedicije.

I konačno igra startuje.

Nude vam se sledeće opcije

Poseta guverneru

VISIT THE GOVERNOR — saznajete ko je s kim u ratu Možete se opredeliti da li ćete voditi svoj privatni rat za račun recimo engleske francuske ili holandske krune (Špance batalite jer su oni toliko bogati da ih treba jedino pljačkati) Poseta guverneru pod određenim uslovima donosi vam unapređenje — ako ste opljačkali neprijateljski grad i pritom postavili guvernera strane koju ste izabrali (najbolje da radite za dve zemlje istovremeno) dolazi do proklamacije i vi dobijete čin Prvi je ENSIGHN potom sledi CAPTAIN MAJOR COLONEL i ADMIRAL. Dalje napredovanje je malo teže — da biste postali COUNT BARUN MARQUIS i konačno DUKE potrebno je da u dva neprijateljska grada postavite guvernera (opisaćemo već kako se to radi)

Guverneri vam obično isplaćuju ucenu za zarobljene gusarske kapetane, setim daju informacije (posle proklamacije) o španskim plemićima koji imaju mape plantaže gde su zatočeni članovi vaše porodice a tu su i guvernerske ćerke koje obavezno imaju udvarača i koje koketiraju sa vama. U početku dok ste praktično golje uglavnom su pomalo i sarkastične na vaš račun ali vremenom kako vi postajete sve bogatiji



Poukova mreža

Spektrum — *Vladimir Janković*

IGRA	POKE	EFEKAT
720	40774 0	Životi
	40360 0	Novac
	37357 0	Ulaznica
ANARCHY	42405,N	N=broj života
ATV SIMULATOR	60250 0	Gorivo
	57318,201	Vreme
BLACK MAGIC	24730 0	Besmrtnost
BRIDE OF FRANKY	40476 201	
	37806,201	Bez neprijatelja
	35486 201	Energija
		Životni eliksir
BUBBLER	57514,12	Životi
	52533,0	Vreme
DEFLEKTOR	34473 0	Životi
	42557,201	Energija
	42827,201	Sprečava Overload
FLASH GORDON	50732 0	Vreme
	49123 0	Municija
NADER NEBULA	61990 0	Životi
I,BALL 2	45392 0	Životi
	63210,0	Vreme
	43618 0	Vreme
KINETIK	61996,0	Životi
MAG MAX	58475 0	Životi
MUTANTS	29667,0	Životi
OUT RUN	40623 0	Vreme

The Treasure Fleet in 1560

Cumana — early October
Puerto Cabello — late October
Maracaibo — early November
Rio de la Hacha — late November
Nombre de Dios — early December
Cartagena — late December
Campeche — late January
Vera Cruz — early February
Havana — early March
Santiago — late March
Florida Channel — late April

The Silver Train in 1560

Cumana — early April
Borburata — late April
Puerto Cabello — early May
Coro — late May
Gibraltar — early June
Maracaibo — late June
Rio de la Hacha — early July
Santa Marta — late July
Cartagena — early August
Panama — late August
Nombre de Dios — early October

The Treasure Fleet in 1600

Cumana — early October
Caracas — late October
Maracaibo — early November
Rio de la Hacha — late November
Santa Marta — early December
Puerto Bello — late December
Cartagena — early January
Campeche — early February
Vera Cruz — late February
Havana — late March
Florida Channel — late April

The Silver Train in 1600

St Thome — early April
Cumana — late April
Caracas — early May
Puerto Cabello — late May
Coro — early June
Gibraltar — late June
Maracaibo — early July
Rio de la Hacha — late July
Santa Marta — early August
Cartagena — late August
Panama — early September
Puerto bello — late October

The Treasure Fleet in 1620

Caracas — early September
Maracaibo — late September
Rio de la Hacha — early October
Santa Marta — late October
Puerto bello — early November
Cartagena — early December
Campeche — early January
Vera Cruz — late January
Havana — late February
Florida Channel — late March

The Silver Train in 1620

St Thome — early March
Cumana — late March
Caracas — early April
Puerto Cabello — late April
Gibraltar — early May
Maracaibo — late May
Rio de la Hacha — early June
Santa Marta — late June
Cartagena — early July
Panama — late July
Puerto bello — early September

The Treasure Fleet in 1660

Caracas — early October
Maracaibo — late October
Rio de la Hacha — early November
Santa Marta — late November
Puerto bello — early December
Cartagena — early January
Campeche — early February
Vera Cruz — late February
Havana — late March
Florida Channel — late April

The Silver Train in 1640

Cumana — early April
Caracas — late April
Gibraltar — early May
Maracaibo — late May
Rio de la Hacha — early June
Santa Marta — early July
Cartagena — late July
Panama — late August
Puerto Bello — early October

njihove erca melekaju i pristaju da vam daju informacije iz kuće njihovog oca — one se uglavnom odnose na srebrni voz i tresor flotu i ne moraju uvek da budu tačne.

Naravno, možete se njima i oženiti, ali i tu obavezno sledi duel sa udvaračem-suparnikom. Ako se i to uspešno okonča stvar u svoje ruke preuzima pop. Ženidbu vam ne preporučujemo jer mreža ženskih informatora daleko više vredi od jedine jedine i to još ko zna kakve što samo vreme može da pokaže.

Odlazak u krčmu

Sledeća opcija je **VISIT TAVERN.**

Tamo možete da nabavljate posadu kupujete mape zakopanog blaga i informacije o gradovima o kojima nema podataka u opciji CITIES (biće govora i o tome) i posebno su važne vesti u taverni.

Mape zakopanog blaga kupujte obavezne — (imate mapu u prilogu) i ako ste upamtili teren lako ćete ga naći koristeći opciju SEARCH pod uslovom da ste na pravom mestu.

Vesti iz taverne, kao i brodske vesti (u toku igre ćete i na to naići) važne su da znate ko je sa kim u ratu koji grad je napala recimo malarija (smanjuje se broj stanovnika) gde je pronađen novi srebrni rudnik (povećava se bogatstvo grada i sl), ukratko korisne su vam u planiranju pohoda.

Opcija TRADE WITH MERCHANT je jednostavna — trgovcu možete prodavati brodove kod njega ih možete i opravljati (doduše ne uvek, ali ako snimate fajl pa ga ponovo učitate to se obavezno dešava i veoma je korisno) a i trgujete robom koju ste zaplenili.

Tu se snabdevate i hranom — pritom pažljivo vodite računa koliko ljudi imate i na koliko dugu se plovidbu otiskujete. Ako ostanete bez hrane posada vam beži i odnosi plen. Na ovo vas kompjuter blagovremeno upozorava.

Podela plena

DIVIDE UP THE PLUNDER — ili podela plena je veoma važna opcija. Treba biti veoma promišljen i plan podeliti u pravome trenutku. Naš savet je da to učinite kada imate najmanji mogući broj ljudi a veliku dobit. Kako se to postiže objasnićemo malo docnije. Po korišćenju te opcije možete planirati novu ekspediciju (počinjete iz početka ali sa već stečenim kapitalom), ili se povući i videti rejting i preći na novi teži nivo.

Odlazak iz grada

LEAVE TOWN — tu je sve jasno.

Neobično je važna opcija CHECK INFORMATION.

Dobijate novi meni sa nizom mogućnosti.

(nastavak putovanja)

CONTINUE TRAVELS je jasna. PARTY STATUS je izveštaj o trenutnom stanju ekspedicije — neobično je važan moral posade. Ona može biti srećna (HEAPPY) zadovoljna (PLEASED) nezadovoljna (UNHEAPPY) i besna (ANGRY).

Ova dva poslednje statusa su veoma opasna. Nikako ne ulazite u gradove u tom trenutku jer mornari dezertiraju i odnose plen. Time gubite piratske bodove. Besna posada može da se pobuni. Tada se borite sa vođom pobune ali najbolje je da do toga ne dođe. To zavisi samo od vas i načina kako planirate i vodite ekspediciju.

PERSONAL STATUS (lični podaci) pored titule koje imate daje još jednu veoma važnu informaciju — vaše godine i zdravlje. Realno je da igru završite u toku 10 godina jer će vas posle toga vaši oficiri posle podele plena poslati u penziju a onda zbogom more ili sve nanovo!

SHIP'S LOG — ili brodski dnevnik koristi vam kao podsetnik. MAPS daje sliku onih mapa koje posedujete a CITIES vam pokazuje stanje garnizona, broj stanovnika i bogatstvo gradova. Ako nema informacije o nekom gradu one se mogu kupiti u taverni.

TAKE SUN SIGHT — pomaže da odredite severnu geografsku širinu — korisno dok dobro ne naučite mapu posle je nepotrebno.

SAVE GAME — Jasno je šta je.

Otisnimo se na more

I došlo je vreme da se otisnemo na more.

U raznim epohama ima raznih brodova — po nama najbolji su SLOOP FRIGATTE i FAST GALEON. Ako napadnete brod, taktika je jednostavna — pucaj jer svaki pogodak smanjuje broj protivničkih mornara i topova a u pravom času priđi i napadaj. Ako si dosta jači neprijatelj se predaje odnosno spušta zastavu, ako nisi sledi mačevanje. Ako ti ne ide najbolje povuci se na levu stranu ekrana i izađi iz okrša. Ako si vešt možeš se oslobodit svojih mornara da bi posle delio plen (dakle polako ih gubi i polako se povlači ali pazi da te neprijatelj ne savlada). Tu je važan moral koji se na ekranu pojavljuje u rasponu od WILD do PANIC (ovo poslednje je znak bliskog kraja).

Kad zarobiš brod možeš ga potopiti ili zadržati (vodi računa da ti flota može imati najviše osam brodova, pa je bolje čuvati samo one veće, čije je nosivost iznad 100 tona).

Kada zarobiš gusarskog kapetana, u principu je bolje da ga čuvaš za otkup, mada ti on nudi i informacije o trezor floti i srebrnom vozu u zamenu za svoju slobodu. One nisu uvek pouzdane. Kada zarobljene gusare prodaješ guvernerima ponekad je dobro odbiti neku ponudu i čekati bolju priliku.

Kada napadaš gradove — postoje tri mogućnosti.

Ako si nadmoćan i ako grad nema tvrđavu odmah se popni na zidine a odlučiće mačevanje.

Ako si slabiji — bitka se odvija na terenu ispred grada. Najbolje je da jednu grupu ostaviš neprijatelju na vidiku a sa drugom se zaobilazno prišunjaš tvrđavi, i onda opet odlučuje koliko si vešt na maču.

Tada napadaš sa svim ljudima koje imaš.

Treći način je napad na tvrđavu sa mora.

Tvoj brod i tvrđava naizmenično pucaju ti manevrišeš i trudiš se da brodom pristaneš tik uz tvrđavu. Tada si na zidinama i sve je već poznato ali ako pristaneš predaleko tvoji ljudi odbijaju da pešače do utvrđenja. I ovi napadi su prilika da ostaneš bez ljudi da bi delio plen, ali i tu vodi računa ako to radiš, jer svaki poraz odnosi piratske poene.

Kada osvojiš grad ako je broj tvojih ljudi veći od zbora preostalih vojnika i građana u gradu možeš da postaviš guvernera. Dakle, opet treba pažljivo planirati.

Ostala je još porodica.

U svakoj igri (sem ekspedicija) moraš da nađeš četiri člana i to prvo sestru potom oca pa majku i na kraju ujaka. Svako od njih ima mapu sa položajem potopljenog španskog broda sa 100 000 zlatnika. Pošto je plan ogroman jasno je da porodicu treba tražiti iz sve snage. Koristi opciju SEARCH.

Još da kažemo kako postati KING'S ADVISOR.

Potrebno vam je 86 piratskih poena igru potom možete igrati i dalje i uz ovo zvanje dobijate već poznate titule kao što su COUNT BARUN MARQUIS i DUKE ali to zahteva dugotrajno igranje i veliki napor.

Ponekad će vam se učiniti da mapa sa gradovima koju prilažemo nije tačna ima gradova kojih nema u nekoj od igara ili se drugačije zovu. Stvar postaje jasnija kada vam kažemo da su u rasponu od 120 godina koliko vam PIRATES nudi jedni gradovi propadali a drugi se rađali.

I sada je sve u vašim rukama.

Brižljivo planirajte ekspedicije, vodite računa kada ćete i koji grad opljačkati (naravno idite na najbogatije — PANAMA je tu najzahvalnija) a koristite priloženu dokumentaciju da bi u pravi čas zaplenili tresor flotu ili srebrni voz, pazite na status posade promišljeno delite plen i napredujte iz nivoa u nivo.

Ne nama je da vam u ovim toplim mesecima poželimo dobro more!

Đorđe Perović

The Treasure Fleet in 1660

Caracas — early September
Maracaibo — late September
Rio de la Hacha — early October
Santa Marta — late October
Puerto bello — early November
Cartagena — early December
Campeche — early January
Vera Cruz — late January
Havana — late February
Florida Channel — late March

The Silver Train in 1660

Cumana — early March
Caracas — late March
Gibraltar — early April
Maracaibo — late April
Rio de la Hacha — early May
Santa Marta — early June
Cartagena — late June
Panama — late July
Puerto bello — early September

The Treasure Fleet in 1680

Caracas — early October
Rio de la Hacha — late October
Santa Marta — early November
Puerto bello — late November
Cartagena — late December
Campeche — late January
Vera Cruz — early February
Havana — early March
Florida Channel — late April

The Silver Train in 1680

Cumana — early April
Caracas — late April
Maracaibo — late May
Rio de la Hacha — late June
Santa Marta — early July
Cartagena — late July
Panama — late August
Puerto bello — early October

ALTERNATIVE WORLD GAMES

Prvi dio je trka u džakovima. Na početku možete igrati protiv suigrača, a kasnije (kad kompjuter shvati da ste vi jedini takmičar) igrate protiv kompjuterovog predstavnika. Kada pritisnete pucanje na palici, pas dotrčava do vašeg predstavnika i igra počinje tek kada sudija da znak zviskom. Ako ne znate da skačete (padate ili vam džak pada na zemlju) pomoći će vam poslušni pas koji će vas vući. Kroz čitavu utrku prolazite ulicom. Čitavo vrijeme se otvaraju šahtovi koji vam zadaju teškoće. Okružuju vas grafički sjajno urađene zgrade sa vešom koji se suši na štrikovima. U daljini se vidi vrh vulkana koji se dimi što podsjeća na Vezuv a grad u kome se takmičite na Napulj (sjajno zar ne!)

Drugi dio je, zamislite ništa drugo nego bacanje čizme u dalj. Ovaj put se nalazite pokraj Koloseuma u Rimu. Takmičite se vi protiv prijatelja ili brata ili igrate sami predstavljajući obje zemlje — Švedsku i Japan. Imate pravo na tri pokušaja. Kad pokrenete džojstik na desnu stranu pojavi vam se crna traka sa žutim pokazivačem koji vam pokazuje sa kolikom snage ćete baciti čizmu. Pokrenete li džojstik na lijevu stranu pokazivač će vam rasti i kad dođe do maksimuma možete baciti čizmu najdalje. Kada čizma padne na dno iz nje ispadne neka crna mrlja koja kasnije dobija oblik glave malog psa. Na mjestu gdje ste bacili čizmu pojavi se zastava zemlje koju predstavljate a u donjem desnom uglu piše koliko ste metara bacili čizmu. Ponekad se nakon bacanja čizme pojavi pas (onaj iz prvog dijela) i polako je odvuče sa ekrana. U slučaju da bacite kako ne treba, tamo gdje se upisuje rezultat pojavljuje se natpis invalid. Takođe čizmu možete baciti i iznad glave. Tada će pasti na vas i istovremeno će vas zemlja jednostavno progutati.

Treći dio je penjanje uz dva kandelabra. Ovde se izgleda nalazimo u Firenci. Pozadina je živopisna i predstavlja jedno renesansno zdanje. Zanimljivo je što se ovaj put na vrijeme takmiče dvije punačke djevojčice. Čitavo vrijeme nadlijeću ptice odnosno golubovi pismonoše iznad glava takmičarki dok im sa balkona jedna djevojka stalno šalje umilne poljupce.

Četvrti dio ujedno i posljednji je po mom mišljenju ipak najbolji. Cilj vam je da protivnicu zbacite sa gondole u vodu kanala. Pozadina je Venecija. Voda i dva mosta su izvrsno grafički urađeni a o obalama ovog uskog kanala i gondoli na kojoj se odvija takmičenje da i ne govorimo. Čitavo vrijeme vam svira kompozicija Hofmanove priče od Žaka Ofenbaha. Takmičite se tako što vam je cilj da odgurnete sa gondole svoju protivnicu pomoću nečega što liči na dvije crvene školjke. Imate samo dvije vrste udaraca. Prvi udarac je kada džojstik usmjerite na lijevo i pritisnete pucanje. Tada će vam „školjka" krenuti iznad glave i protivnicu udariti u glavu. Druga vrsta udarca je kada džojstik usmjerite prema dole ili gore. Tada će vam „školjke" krenuti iza leđa i udariti protivnicu oko pasa. Potrebno je šest puta zaredom udariti protivnicu i ona će pasti u vodu.

Alternative world games je sigurno inspirisana silnim olimpijadama, ali scenario igre bez sumnje predstavlja šalu, ali i jako uspješnu parodiju na svoje prethodnike.

Dejan Krajačić

Spektrum

SIDE ARMS

STRELJAČKO ORUŽJE



Jednom u sto godina naslov koji odgovara igri

Naoružani i oklopljeni letite sleva udesno i gađate sve što vam se isprečí — pa čak imajte u vidu i sve što vas napada s leđa (jer ovde imate i takvih podmuklica)

Naravno na kraju svakog nivoa čeka vas gadan i maltene neuništiv neprijatelj naravno u početku imate srednje žalosnu pucaljku naravno usput pojačavate vatrenu i odbrambenu moć

Moći ćete da pokupite

— pojačivač energije tj brzine,

— trocevku

— teški laser

— trocevku koja kontinuirano puca

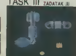
— automatski laser koji kontinuirano puca i

— kuglice koje lete oko vas i služe kao štit

Ako i pri svem tom ne uspete da doakate neprijateljima ne pucajte u programera on je učinio sve što je mogao.

Komodor

TASK III ZADATAK III



Smisliti originalnu pucačku igru postalo je veoma teško Zato se mnogi i ne trude

Ako zanemarite scenario koji kaže da je godina 8038 da je zli car iz neke galaksije u čestregiji porobio itd itd ako dakle stvari pogledate trezveno videćete igru u šesnaest nivoa vertikalno skrolovanu Posle šesnaestog nivoa kad čeka vas dramatičan obračun sa glavnim zlikovcem

Vreme vam je ograničeno ali količina neprijatelja nije Brze su i gadne tuđinske letelice ali i vi konja sa trku imate (ukoliko ova metafora može opstati u visokotehnološkom raju) Programer je smislio trostruki laserski mitraljez dopunske laserske oruđe sa strane i u repu „pametne bombe" zaštitni satelit sredstvo za usporavanje neprijatelja i nešto veoma lepo nazvano **Emergency Package** ili paket za hitne slučajeve pritisnete RUN/STOP i dobijete sav dopunski arsenal

No ipak je najbolje sačekati da neko provali pouk sa besmrtnost

BLOOD VALLEY

KRVAVA DOLINA



Nije ovo puka fraza dolina će zaista biti krvava (bar delimično) ako vas uhvate jer vi ste rob koga hoće da prinesu na žrtvu Naravno očiro ste osudili taj varvarski običaj ali zaostala sredina nije vas razumela stoga morate nastojati da likvidirate vladara i tri njegova doglavnika Da biste to izveli morate obilaziti noga po svakojakim predelima boriti se sa gomilom neprijatelja i steći što više možete hrane zlata i čini

Možete igrati zamrlii udvoje u ovom drugom slučaju jedan igrač biva gonilac a drugi progonjeni Ekran se deli nadvoje i kom opanci kam oboje Gonilac ima prednost utoliko što svoje ili doglavnike može razmestiti manje-više gde hoće (Doduše niko mu ne garantuje da će progonjeni naići baš na tu lokaciju)

No pre no što išta uradite morate odlučiti šta da budete varvarin sveštenik ili lopov Zadaci tokom igre malo će se menjati zavisno od vaše uloge ali glavni principi ostaju isti Ne manje se ni rok u svakoj varijanti imate pet dana da obavite svoje poslove Ako preživite vikend je vaš

Atari ST

OIDS

OIDI



Počela je u naučnoj fantastici reču androidi koja označava veštački stvoreno biće koliko biološko toliko i kibernetičko veoma slično čoveku U kompjuterskim igrama za njim su došli droidi a sad su kako vidimo otkriveni i oidi

Čija će sudbina biti gora od smrti ako ih ne izbavite Zato ste krenuli na njihove planete da ih pokupite Letenje je podnošljivo ali kad se približite planeti nećete više znati gde vam je glava. Kao prvo morate paziti ali teže da se ne biste jednostavno simeknuli na tle i zaginuli Kao drugo gospodari oida opremljeni su svim i svačim nevidljivim poljima sila oko zgrada protivavionskom artiljerijom projektilima koji vas gone u stopu moraćete biti itekako brzi i vešti da biste ostali živi i čitavi

Ometaće vas čak i oidi uspaničeni oni će trčkarati gore-dole i lako se često naći baš onde gde treba da sletite

Ako preživite sve te zavrzlame steći ćete neku vrstu poklona moći ćete da poučeni iskustvom dizajnirate svoju ličnu varijantu sve igre U njoj bi trebalo da budete nepobedivi

STREETFIGHTER

ULIČNI BORAC



U početku beše RENEGADE i pošto je išao k'o alva našao se neko ko je primenio Vukovu formulu isto to samo malo drukčije i napravio STREETFIGHTER a.

Isto to znači da se i ovde sva vreme tučete a malo drukčije znači da su sprajtovi veći i bolje animirani i da je scenario izmenjen U novoj varijanti putujete kroz pet zemalja (Japan SAD Englesku Kinu i Tajland) i u svakoj imate posla sa izabranim predstavnikom lokalnih siledžija

U međuigrama ili bonus nivoima vežbate karate golom rukom razbijate cigle

Naravno svaki protivnik ima i neka rudimenta ličnog stila Englez na primer rado udara glavom jedan Japanac (nindža) pojavljuje se i nestaje Tajlanđanin ima duge ruke itd

Udarci su manje-više kao i u RENEGADE-u a i taktika najbolja će vam biti ako se posle svakog udarca povučete Bežanova majka ne plače

PACLAND

PEKMENOVA DOMOVINA



Nostalgija čini svoje sve je više novih igara koje se vraćaju motivima iz starih Pekmen okruglir čovečuljak dosjen likova iz kompjuterskih igara ponovo je na sceni razume se i on i scena postali su komplikovaniji i grafički bogatiji

Dok putuje kroz svoju domovinu (1 putovanje = 4 nivoa) čovečuljak nailazi na svoje verne neprijatelje duhove Oni su se u međuvremenu veoma prozlili jure za njim gaze ga automobilima gađaju ga raznim predmetima s prozora ganjaju ga avionima itd Ili pelm ni to nije dosta nego mu još postavljaju i zamke.

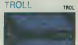
Sirotí Pekmen mora stalno da ih izvrdava što nije lako ni u običnim prilikama akamoli (recimo) onda kao prelazi reku skačući s jednog pokretnog balvana na drugi

Na sreću povremano nailazi na one enaigetske tablete (sećate li se prve verzije?) i kako koju proguta tako sliče (privremenu) moć da jede duhove Našlo rede ima priliku i da se domogne šešira koji mu daje nepobedivost i letećih čizama vrlo zgodnih za preskakanje prepreka

Pored toga računa i s vašom igračkom veštinom

TROLL

TROL



Teško vama

Naći ćete se u podzemnom svetu okruženi zledusima zidevima i kristalima u svetu podeljenom nadvoje Zledusi će podizati zidove oko vas i tako vas prilično efikasno izolirati među kristalima moraćete naći kristalni ključ i potom ga uglaviti na pravo mesto u amuletu koristeći se piramidalnim predmetima prelazićete iz gornje u donju polovinu sveta (i obratno) iz nivoa u nivo prolazićete kroz rupe koje verovali ili ne nosite sa sobom Izvadite rupu bacite je na tlo i gotov posao To jest bio bi gotov posao kad biste znali je li rupa dobroćudna ili ne a pi redu rupe određuje mašina slična „jednorukom Džeku" (tj sudbinu vam određuje slučaj) Da bi se stvar još više zamrsila postoje i lutajuće rupe pogibeljne za vas Nekad su odmah vidljive a nekad ih kriju zidovi (oni zidovi koje kao što rekosmo podižu vaši neprijatelji) Uostalom i mnogi predmeti su tako sakriveni

Prebiće vas tri znoja (najmanje) dok ne pokupite sve ključeve i ne vratite se u svoj svet A kad se vratite nadam se da ćete otvarati četvore oči da ne biste opet prepali u ovo opako podzemlje.

PINK PANTHER

PANTER PANTA



Jasta stigao je i on Posle svih onih ljudi i životinja iz crtaća morao je doći na red i naš prijatelj Panta.

Koji ima jednostavan plan zaposliti se kao batler kod milionera maznuti sve moguće dragocenosti a onda zalmiti i utaditi to isto sledećem milioneru Kad popali i četvrtog imaće dovoljno para da se povuče na ostrvo svojih snova — verovatno blizu onoga na kome se već odmara krt Menli

Panta dakle kreće na posao Prva je prepreka cilindar u agenciji za zapošljavanje batlera saznaje da ga mora imati kao deo odora stoga mora ići u radnju da ga kupi i to je dosadni deo igre

Uzbudljivi deo počinje u milionarovoj kući Panta se noću šunja i krade ali gazda je mesečar pa se šunja i on. Treba ga dakle voditi gore-dole da ne bi naletao ni na šta (tj da se ne bi probudio i onda uhvatio lopova) Panta se to koristi zvono (na čiji zvuk gazda menja pravac kretanja) daske (za prelaženje preko prepreka) i figurice na naduvavanja (da zavara trag)

Znam da nije lepo krasti ali Panta mi je izuzetno drag lik i želim mu uspeh u ovom poduhvatu

TIM 600

beskrajni prostori RAČUNARSTVA

- **TIM 600 JE VIŠEKORISNIČKI SUPER MIKRORAČUNARSKI SISTEM VISOKIH PERFORMANSI, KOJI KRUNIŠE VLASTITI RAZVOJNI PUT U RAČUNARSTVU INSTITUTA „MIHAJLO PUPIN"**
- **NOVI SNAŽNI 32-BITNI SISTEM JE OD SVOJIH PRETHODNIKA ZADRŽAO VISOKU POUZDANOST U RADU.**
- **TIM 600 JE ZASNOVAN NA NAJPOPULARNIJIM MIKROPROCESORIMA INTEL 80386/80387 I NA SVETSKOM INDUSTRIJSKOM STANDARDU OPERATIVNOG SISTEMA UNIX SYSTEM V.3.**
- **ARHITEKTURA SISTEMA TIM 600 OSLANJA SE NA TRI MAGISTRALE ZA PRENOS PODATAKA IZMEĐU SISTEMSKIH RESURSA: 32-BITNA SISTEMSKA CP-MEM MAGISTRALA, 16-BITNA MAGISTRALA ZA U/I AKTIVNOSTI; 8-BITNA MAGISTRALA ZA SCSI UREĐAJE.**
- **PRIMENA SISTEMA TIM-600 JE VRLO ŠIROKA: ZA PROCESIRANJE INFORMACIJA, UPRAVLJANJE PROCESIMA I AUTOMATIZACIJU POSLOVANJA (POSLOVNE OBRADE, INDUSTRIJA, NAUKA, TEHNIKA, ZDRAVSTVO, OBRAZOVANJE, BANKE, TRGOVINE, DRUŠTVENE SLUŽBE I JAVNA UPRAVA).**



• TEHNIČKE KARAKTERISTIKE

— CPU (32-bita 80386, 20 MHz, 5 miliona operacija u sekundi)
— MEMORIJA (8 MB sa mogućnošću proširanja u modulima od 2MB).
— RAČUNARSKE KOMUNIKACIJE protokoli BSC, SNA/SDLC ili X 25
— RAČUNARSKA GRAFIKA visoka rezolucije 1024×1024 sa 256 boja
— PROCESOR MASOVNIH MAGNETNIH MEDIJA sa paritarnim kontrolarima za diskete, fiksna diskova i kasatne traka
— PROGRAMSKI JEZICI C, COBOL, FORTRAN, PASCAL, BASIC i drugi.
— SAVREMENI SOFTVERSKI ALATI za razvoj, projaktovanja i održavanja softvara IV ganaracija, UPRAVLJANJE BAZAMA PODATAKA i drugo (korišćanja ORACLE, INFORMIX itd)
— DISTRIBUIRANA OBRADA I POVEZIVANJE SA PERSONALNIM ILI SA DRUGIM VELIKIM RAČUNARIMA. Maksimalni broj aktivnih korisnika koja sistam podržava zavisi od tipa primana i izabrane konfiguracija (standardno do 32 korisnika, proširanja do 64 max.)
— Nova programska podrška za rad u mraži TIMNET®
— Napajanja 220 V, 50 Hz, prakidački izvor 360 VA
— Dimanzija računarskog kabineta 680×235×710 mm

IMP — RJ RAČUNARSTVO
tel. 772-876 telex 11584 yu imp bg
talafax 011/774614

Institut MIHAJLO PUPIN

PC810

UNIS

ELEKTRONIKA
TELEKOMUNIKACIJE
INFORMATIKA



PC 810 je reprezentativni model nove tročlane UNIS — NCR serije personalnih računara

PC 710
PC 810
PC 916

Odlikuje se tzv. Incremental Workstation Architecture koje omogućava korišćenje najnovije tehnologije pojednostavljuje servisiranje i pruža mogućnost proširivanja počešnih do veoma moćnih konfiguracija.

Uzimajući sve prednosti dizajna integrisanih ploča (VLSI) i visoku tehnološku opremljenost, PC 810 nudi pojačane AT-učinke. Šest AT -kompatibilnih slotova za proširenje pružaju mogućnost izbora različite grafike i konfigurisanje različitih memorijskih medija i čine PC 810 snažnim i dovoljno fleksibilnim za većinu korisničkih zahtjeva.
U PC 810 je na osnovnoj procesorskoj ploči ugrađen 16-bitni INTEL 80286 mikroprocesor koji funkcioniše dvojnom brzinom od 8 ili 10 MHz. Na istoj ploči nalaze se i podnožje za ugradnju aritmetičkog koprocesora 80287 ROM-čipe za zaštitu za korisnika specifičnog softvera kao i 640 KB RAM memorije. Korištenjem ploča za proširenje može se RAM memorije proširiti i do 16 MB.

OPERATIVNI SISTEM

— PC kompatibilan NCR-DOS omogućava korištenje široke palete profesionalnog softvera
— XENIX (opcionalno) je multi user multi-tasking operativni sistem opšte namjene. Prilagođen je personalnim računarima čija je osnova INTEL ov mikroprocesor 80286.
Kod PC 810 je predviđena i mogućnost ugradnje INTEL 80386 mikroprocesora, čime ovaj model prerasta u najsavršeniji iz nove serije — PC 916

KOMUNIKACIONE MOGUĆNOSTI

PC 810 podržava sve standardne komunikacione protokole i može se vezati u mrežu sa proizvodima svih u svijetu priznatih proizvođača računarske opreme

Tel. 071/205-455
Telex 41570/52
71000 Sarajevo
Tršćanska 7

TABELA SA OSNOVNIM KARAKTERISTIKAMA:

CENTRALNI PROCESOR	INTEL 80286 (opcionalno 80386)
KOPROCESOR	INTEL 80287
BRZINA RADA	6/10 MHZ
GLAVNA MEMORIJA	640 KB — 16 MB
DISKETNA JEDINICA 5 1/4"	1,2 MB
DISKETNA JEDINICA 3 5"	2×720 KB ili 1 44 MB
WINCHESTER DISKOVI	2×20/30/44/70/115 MB
STREAMER TRAKA	40 ili 60 MB
GRAFIČKE KARTICE	CGA, EGA i VGA
EKRANI	15" Monohrom, 14" Kolor
REZOLUCIJA EKRANA	MONO 640×(200×2) KOLOR 640×200
INTERFEJSI	RS 232, CENTRONICS
RADNI USLOVI — Temperatura	10—32°C
— Rel. vlažnost	20—80%
OPERATIVNI SISTEM	NCR DOS, XENIX, predviđen OS/2
KOMUNIKACIJE — Protokoli	TTY, HOLC, SOLC 3270/3770 BSC, 2780/3780 BSC
— Mreže	SNA, X 25, TOKEN RING
KOMPATIBILNOST	AT i PS/2 KOMPATIBILAN



OOUR „MARKETING"
Sektor plasmana opreme za AOP